UNIVERSITY OF SOUTHAMPTON

Multi-Agent Negotiation using Trust and Persuasion

by

Sarvapali Dyanand Ramchurn

A thesis submitted in partial fulfillment for the degree of Doctor of Philosophy

in the
Faculty of Engineering and Applied Science
School of Electronics and Computer Science

December 2004

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND APPLIED SCIENCE
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by Sarvapali Dyanand Ramchurn

In this thesis, we propose a panoply of tools and techniques to manage inter-agent dependencies in open, distributed multi-agent systems that have significant degrees of uncertainty. In particular, we focus on situations in which agents are involved in repeated interactions where they need to negotiate to resolve conflicts that may arise between them. To this end, we endow agents with decision making models that exploit the notion of trust and use persuasive techniques during the negotiation process to reduce the level of uncertainty and achieve better deals in the long run.

Firstly, we develop and evaluate a new trust model (called CREDIT) that allows agents to measure the degree of trust they should place in their opponents. This model reduces the uncertainty that agents have about their opponents' reliability. Thus, over repeated interactions, CREDIT enables agents to model their opponents' reliability using probabilistic techniques and a fuzzy reasoning mechanism that allows the combination of measures based on reputation (indirect interactions) and confidence (direct interactions). In so doing, CREDIT takes a wider range of behaviour-influencing factors into account than existing models, including the norms of the agents and the institution within which transactions occur. We then explore a novel application of trust models by showing how the measures developed in CREDIT ca be applied negotiations in multiple encounters. Specifically we show that agents that use CREDIT are able to avoid unreliable agents, both during the selection of interaction partners and during the negotiation process itself by using trust to adjust their negotiation stance. Also, we empirically show that agents are able to reach good deals with agents that are unreliable to some degree (rather than completely unreliable) and with those that try to strategically exploit their opponent.

Secondly, having applied CREDIT to negotiations, we further extend the application of trust to reduce uncertainty about the reliability of agents in mechanism design (where the honesty of agents is elicited by the protocol). Thus, we develop Trust-Based Mechanism Design (TBMD) that allows agents using a trust model (such as CREDIT) to reach efficient agreements that choose the most reliable agents in the long run. In particular, we show that our mechanism enforces truth-telling from the agents (i.e. it is incentive

compatible), both about their perceived reliability of their opponent and their valuations for the goods to be traded. In proving the latter properties, our trust-based mechanism is shown to be the first reputation mechanism that implements individual rationality, incentive compatibility, and efficiency. Our trust-based mechanism is also empirically evaluated and shown to be better than other comparable models in reaching the outcome that maximises all the negotiating agents' utilities and in choosing the most reliable agents in the long run.

Thirdly, having explored ways to reduce uncertainties about reliability and honesty, we use persuasive negotiation techniques to tackle issues associated with uncertainties that agents have about the preferences and the space of possible agreements. To this end, we propose a novel protocol and reasoning mechanism that agents can use to generate and evaluate persuasive elements, such as promises of future rewards, to support the offers they make during negotiation. These persuasive elements aim to make offers more attractive over multiple encounters given the absence of information about an opponent's discount factors or exact payoffs. Specifically, we empirically demonstrate that agents are able to achieve a larger number of agreements and a higher expected utility over repeated encounters when they are given the capability to give or ask for rewards. Moreover, we develop a novel strategy using this protocol and show that it outperforms existing state of the art heuristic negotiation models.

Finally, the applicability of persuasive negotiation and CREDIT is exemplified through a practical implementation in a pervasive computing environment. In this context, the negotiation mechanism is implemented in an instant messaging platform (JABBER) and used to resolve conflicts between group and individual preferences that arise in a meeting room scenario. In particular, we show how persuasive negotiation and trust permit a flexible management of interruptions by allowing intrusions to happen at appropriate times during the meeting while still managing to satisfy the preferences of all parties present.

Contents

List of Figures

List of Acronyms

ABN Argumentation-Based Negotiation

CREDIT **C**onfidence and **RE**putation **D**efining **I**nteraction-based **T**rust

FIPA ACL Foundation for Intelligent Physical Agents' Agent Communication Language

IMMPD Iterated Multi-Move Prisoner's Dilemma

KQML Knowledge Query and Manipulation Language

MAS Multi-Agent Systems

MD Mechanism Design

MMPD Multi-Move Prisoner's Dilemma

PD Prisoner's Dilemma

PN Persuasive Negotiation

POS probability of success

RBT Reward Based Tactic

RWG Reward Generation Mechanism

TBM Trust-Based Mechanism

TBMD Trust-Based Mechanism Design

VCG Vickrey-Clarkes-Groves

Acknowledgements

I would first like to thank the people who have helped me write up this PhD. In particular, I thank my supervisors (and friends), Prof. Nicholas R. Jennings and Prof. Carles Sierra for their unrelenting support and guidance throughout the three years of my PhD, Prof. Lluis Godo who have been extremely supportive supervisors and co-authors of the main works I have been engaged. Also, I would like to thank my friends Raj Dash, with whom chapter 6 was co-written, and Iyad Rahwan with whom chapter 2 was written. My thanks also go to Cora Excelente Toledo, Ronald Ashri, Steve Munroe, Vijay Dialani, and Krishnen Vytelingum for bearing with my complaints and providing me with advice during these three years. I also thank my other co-authors for their useful discussions and contributions to parts of this thesis, namely Pietro Panzarasa and Dong Huynh for the review on trust, Iyad Rahwan, Simon Parsons, Peter McBurney, and Liz Sonenberg for the review on argumentation based negotiation, and David De Roure, Benjamin Deitch, Mark Thompson and Michael Luck for our work on applying negotiation in pervasive computing environments.

This work has also been possible thanks to the many friends I have, in Southampton and elsewhere, who have contributed in one way or another to a rich and less stressful life in UK (and abroad) and helped me enjoy the little free time this PhD allowed me. In this respect, I would like to thank Khalid, Vanessa, Ouma, Zarine, Lu, Ioanna, Stergios, Tim, Ibrahim, Sarah, and Joanne.

Last, but not least, I am most grateful to my parents and brothers for their blessings, love, affection, and interest in my work. Their moral support and timely advice have been instrumental to my life and, in particular, the completion of this thesis.

To Dadi

Chapter 1

Introduction

Open distributed computing applications are becoming increasingly commonplace in our society. In most cases, these applications are composed of multiple actors or agents, each with its own aims and objectives. In such complex systems, dependencies between these multiple agents are inevitable, and generally speaking, they cannot all be predicted in advance. Therefore a runtime mechanism is needed to manage them and to resolve any conflicts that might ensue in a context-dependent manner. We believe the de facto mechanism for achieving this is *automated negotiation* and this is the area explored in this thesis.

However, designing effective negotiation mechanisms for open distributed applications is a major research challenge. Specifically, there is a high degree of uncertainty in the variables that impact on negotiations. This is because the actions of the actors (i.e. what they are able to achieve), their preferences (i.e. what outcomes they deem possible and would prefer), their honesty (i.e. to what extent they want to reveal private information truthfully), and their reliability (i.e. how good they are at what they say they can do) are not public knowledge. This uncertainty may, in turn, prevent the agents from reaching good agreements during negotiations (because they are not able to make decisions with full knowledge of the effects of their actions). Given this, the underlying motivation of this thesis is to devise techniques to reduce this uncertainty so that agents can reach better agreements through automated negotiation. In particular, this involves modelling the variables that are prone to uncertainty using decision theoretic techniques (e.g. statistics and/or fuzzy reasoning), determining ways in which the output of such techniques can be used in automated negotiation, and detailing how this output can be refined over multiple encounters between the agents in order to make the search for the best agreement quicker. Against this background, we develop three general classes of techniques that aim to enhance the outcome of such repeated encounters. First, we propose that agents model their opponents' reliability through the notion of trust. To this end, we develop the CREDIT[1] trust model. Using CREDIT, agents are able to adapt their negotiation stance in bargaining encounters according to how trustworthy (reliable and honest) they believe their opponent to be in enacting the contents of a contract. Second, we develop the notion of Trust-Based Mechanism Design (TBMD) that uses game theoretic techniques to select the most reliable agents in the system by incentivizing them to honestly reveal their preferences and their trustworthiness. Third, we develop a novel mechanism for Persuasive Negotiation (PN) for reducing the uncertainty in repeated encounters by allowing agents to constrain the space of outcomes that they need to search in order to find an agreement. Thus, in persuasive negotiation, agents can ask for or give rewards, which constrain future encounters, in an attempt to make an offer in the current negotiation more acceptable.

The rest of this chapter is structured as follows. Section 1.1 maps out the general need for automated negotiation in Multi-Agent Systems (MAS). In section 1.2 we discuss the techniques that are used in negotiation and identify those attributes of negotiation encounters that can be uncertain. In section 1.3, we then discuss the issue of trust as a means to reduce the uncertainty about the honesty or reliability of agents. Then, in section 1.4 we discuss how uncertainties about the action set and preferences of agents can be dealt with in persuasive negotiation. The aims and objectives, as well as the main contributions of the thesis, are outlined in section 1.5 and the structure of the remainder of this thesis is given in section 1.6.

1.1 Motivation for Research

Many computer applications are open distributed systems in which the (very many) constituent components are spread throughout a network, in a decentralised control regime, and are subject to constant change throughout the system's lifetime. Examples include the Grid (Foster and Kesselman, 1998), peer-to-peer computing (Ripeanu et al., 2002), the semantic web (Berners-Lee et al., 2001), web services (Seth, 2003), e-business (Kersten and Lo, 2001), m-commerce (Tveit, 2001; Vulkan, 1999), autonomic computing (Kephart and Chess, 2003), and pervasive computing environments (Satyanarayanan, 2001). Such open distributed systems are typically composed of various stakeholders, each with their own, possibly conflicting, interests. Therefore, there is a need to have autonomous components, that represent these stakeholders, and act and interact in flexible ways in order to achieve their design objectives in uncertain and dynamic environments (Simon, 1996). Given this, agent based computing has been advocated as the natural computation model for such systems (Jennings, 2001).

More specifically, the agent paradigm allows the decomposition of large, complex, and distributed systems into a number of autonomous entities that can interact with each

[1] **C**onfidence and **RE**putation **D**efining **I**nteraction-based **T**rust (CREDIT).

other in order to achieve their individual objectives (Jennings, 2000). To be even more precise, the following definition of an *agent* will be used throughout this work:

Definition 1.1. An agent is a computer system *situated* in an *environment*, and capable of *flexible autonomous action* in this environment in order to meet its *design objectives* (adapted from Wooldridge and Jennings (1995)).

This definition highlights the fact that an agent must have the following properties:

- Reactivity — the ability to respond to changes to its perceived environment including those changes that result from the actions of other agents.
- Proactiveness — the ability to exploit opportunities to satisfy its goals, rather than constraining itself to predefined rules.
- Social ability — the ability to interact with other agents in its environment to satisfy its goals.

The last of these properties is probably the main defining characteristics of agents that are situated in MAS. In this work, agents within such systems are assured to interact with one another according to some *interaction mechanism* that guides the participants to a particular outcome:

Definition 1.2. An interaction mechanism is a means by which agents are able to achieve one or more of the following: (i) exchange information, (ii) coordinate their actions and (iii) resolve their conflicts.

Given this, open distributed systems can be modelled as open multi-agent systems that are composed of autonomous agents that interact with one another using particular interaction mechanisms. Obviously, depending on the nature of the interaction, different types of interaction mechanisms will be used. Broadly speaking, we can characterise the nature of interactions in the following ways:

- *Competitive interactions* — agents interact to satisfy their *own* preferences. These preferences are usually captured through their *utility function* which assigns a score (usually a real value) to particular outcomes in the interaction. In such competitive interactions, agents try to maximise their utility function and are hence termed *selfish* or *self-interested*. Specifically, the agents try to deduce the course of action that maximises their utility given their knowledge of their environment and the possible actions of other agents. This may involve hiding their preferences since doing otherwise might lead to a low utility deal being achieved.[2] Given this, MAS

[2]Such decision making based on the computation of the utility maximising action relative to other agents' actions is normally termed *strategic* decision making (Rosenschein and Zlotkin, 1994).

designers have to engineer the system that guides such competitive interactions through *protocols* so that agents do not unduly exploit one another or the overall system in seeking to maximise their individual utility. In so doing, the designer can ensure that the system is fair and incentivises individual stakeholders to participate in it. Generally speaking, these protocols dictate the range of actions that agents can perform (i.e. their action set), the sequence of actions that are permissible (e.g. each agent performing only one action concurrently with others or a number of actions sequentially with others' actions), and how the agents' actions translate into an outcome (Dash et al., 2003; Rosenschein and Zlotkin, 1994; Sandholm, 1999). Given the system's protocols, the agents' owners need to define the *strategy* of the agents that can achieve their goals (i.e. given the history of actions, what an agent is supposed to do next).

- *Cooperative interactions* — agents interact in order to try and maximise the sum of all their utilities (also termed the *social welfare* (Mas-Colell et al., 1995)) (as opposed to their individual utility in the competitive case). In these interactions, agents totally devote themselves to the group's cause even at the expense of their individual goals (Pynadath and Tambe, 2002) (i.e. even if their individual utility is low in the chosen outcome). In this context, the main problem of the MAS designer is that of devising algorithms (i.e. covering both the protocol used and the strategy of the individual agents) that can find a globally optimum set of actions that still manage to satisfy each agent's constraints (Yokoo and Hirayama, 2000; Becker et al., 2003). The problem of finding the optimum set of actions is usually exacerbated in this case by uncertainties in the knowledge agents have about each other's actions and the number of constraints (or variables) that exist for each agent.

In this thesis we focus on interaction mechanisms that deal with competitive interactions since this represents the most general class of interactions (i.e. a competitive interaction can be reduced to a cooperative one by changing the nature of the utility function of each agent). In particular, as stated earlier, agents, while having selfish interests, may need to collaborate to achieve their goals. In such contexts, agents usually aim to find an *agreement* that determines a course of action that maximises their individual utilities. To this end, a number of techniques have been devised, forming the general class of *negotiation* mechanisms, more commonly known as *automated negotiation* mechanisms in the MAS literature.

1.2 Automated Negotiation Mechanisms

Negotiation has been defined in many different ways (see (Walton and Krabbe, 1995; Fisher and Ury, 1983; Rosenschein and Zlotkin, 1994; Jennings et al., 2000)). However, fundamentally, its main goal is to achieve an *agreement* over some *issue(s)* of contention. In this thesis we adopt the following definition:

Definition 1.3. Negotiation is an interaction mechanism that aims to resolve a *conflict of interest* between two or more parties through the use of a defined *protocol* and the strategies of the agents (adapted from (Jennings et al., 2001)).

The protocol usually determines the sequence of steps agents need to follow during negotiation, while the agents' strategies are part of their reasoning mechanism (which also involves information gathering and analysis, and offer generation components). As can be deduced from the above definition, the *aim* of negotiation is to find an agreement that satisfies the agents' preferences or constraints, but such encounters do not always end up in an agreement (and agents may gain zero or negative utility from this). Non-agreement can happen as a result of a lack of time, an unavailability of viable options for the participants (that could result from a lack of knowledge about the participants' preferences), or an incompatibility between the strategies used by the agents (Fisher and Ury, 1983; Raiffa, 1982). However, if an agreement is feasible and the agents are actually able to achieve it, all parties are normally *committed* to enacting the contents of the agreement (Jennings, 1993). In this work, we define a commitment as follows:

Definition 1.4. A commitment is a pledge by an agent to ensure that the contents of the commitments are achieved through some actions (adapted from (Jennings, 1993)).

The properties of the agreement reached (i.e. the type of actions agents commit themselves to) are dictated by the negotiation mechanism used (i.e. the protocol and strategies of agents). For example, if the mechanism allows agents to exhaustively explore the space of all possible agreements, the agreement chosen should be one that maximises all negotiating agents' utilities. In contrast, if the negotiation mechanism only allows an agent to accept or reject only one offer (e.g. in take it or leave it negotiation), the agreement may not be the most efficient one that could be obtained. Moreover, the type of mechanism chosen by the system designer may, in turn, depend on a number of factors, among which we note the following:

- The context of application — while some applications give an upper hand to the system designer to formulate a protocol that meets certain criteria (wanted by the designer), other applications may give more control to the individual agents' owners. For example, in selling licenses for bandwidth to telecommunication companies, a government agency (the system designer) may decide on a particular protocol that the companies need to comply with in placing their offers and, in so doing, elicits their true preferences and maximises the agency's profit (Krishna, 2002). On the other hand, traders in a stock market have to decide on their own (negotiation) strategies in order to get the best profit in the system given the rules that are in place.

- The *uncertainty* prevailing in the application — in most applications negotiations have to take place in an environment where there is a degree of uncertainty. In this context, uncertainty about a particular property or attribute means that there is a lack of information about that property or attribute and there is no statistical model for this. For example, agents may be uncertain about their exact preferences or about the actions they can perform in the environment. Agents may also be uncertain about their opponents' reliability (i.e. how good they are at doing what they say they can do) and their honesty (i.e. whether they tell the truth about the information have). In such cases, the protocol and the agents' strategies used for the negotiation will have to take these into account if the agents are to come to acceptable outcomes. Such uncertainties can be reduced in a number of ways including, but not limited to:

  - Developing decision making models that allow agents to model those attributes or properties liable to uncertainty. In such contexts, we expect agents to use decision theoretic techniques such as statistics (Savage, 1954) or fuzzy reasoning (Zadeh, 1965; Mamdani, 1977) that permit such a modelling.
  - Adapting the protocol to permit agents to reduce the number of variables over which the uncertainty applies. This may involve using a protocol that forces the agents to reveal all the information available to each of them (Krishna, 2002) or constraining the number of actions that they may perform (Hovi, 1998; Mas-Colell et al., 1995),.

Given this, a number of automated negotiation mechanisms have been devised to cater for different contexts and uncertainties. We can broadly classify these into following categories (see figure 1.1):

- Bargaining — this typically involves the exchange of offers between the interacting agents until an agreement is reached (this is often termed 'negotiation' in some cases (Jennings, 2001; Faratin et al., 1998)). In this context, each offer implies a conditional commitment on the part of the sending agent that it will enact the contents of the offer if and only if the recipient sends an 'agree' message. The contents of the offer or the negotiation object can vary from the very simple (e.g. based on price or quality only) to the extremely complex (e.g. involving trade-offs between price and quality) (Klein et al., 2003; Faratin et al., 2002). The negotiation object may also be dynamically changed by adding other issues during the negotiation process or by constraints imposed during other (concurrent or previous) negotiation encounters (games).

  Bargaining is appealing in situations where it is not possible to have a central authority that can generate an outcome that maximises the utility of all interacting agents. Also, bargaining protocols do not usually assume known preferences,

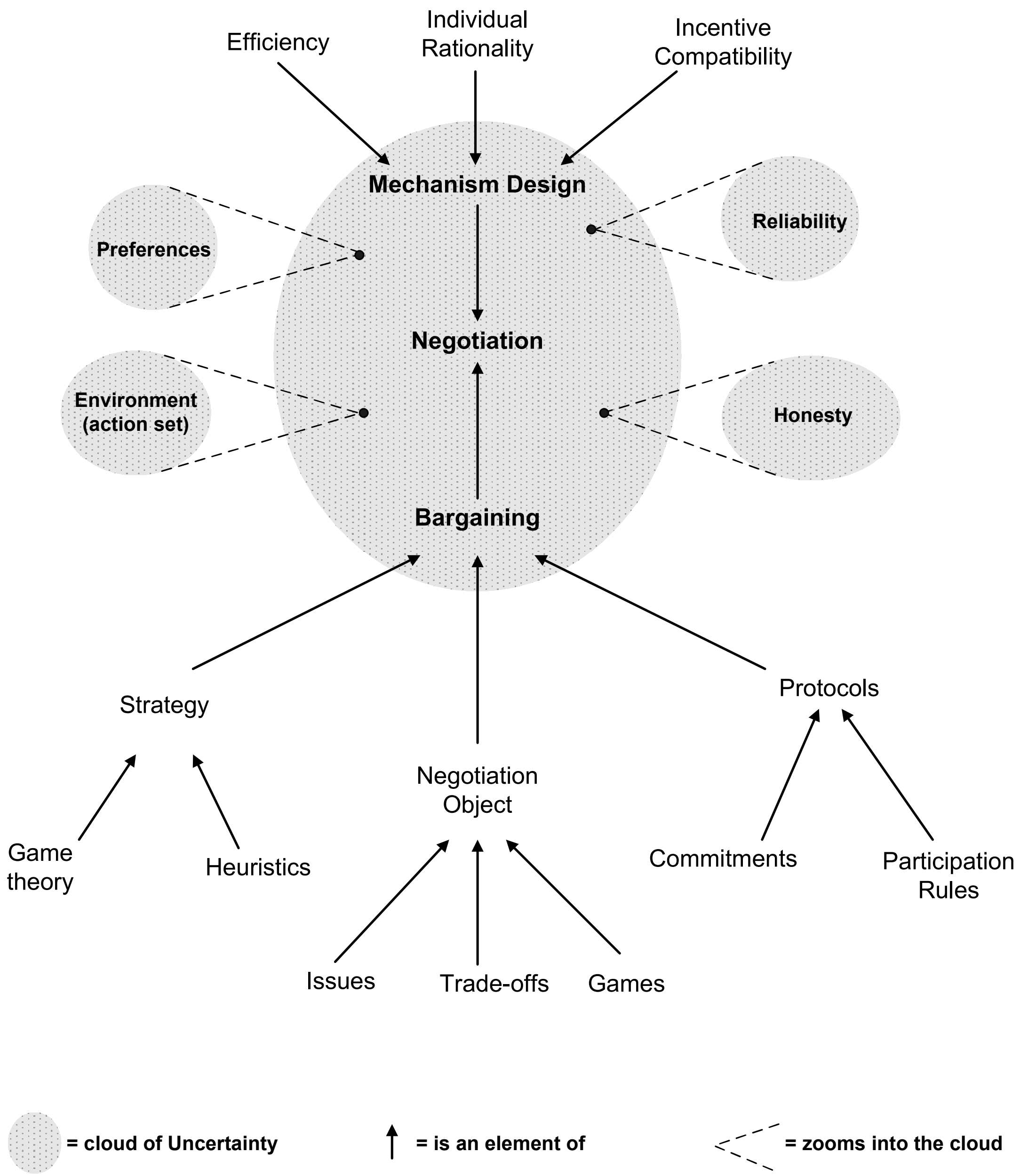


FIGURE 1.1: Approaches to negotiation in multi-agent systems and the cloud of uncertainty covering various aspects of the interaction.

reliability levels, action sets, or degree of honesty of the agents. Typically they only impose the sequence of exchange of offers (e.g. alternating offers or 'take it or leave it') or the participation rules that determine when agents are allowed to leave the negotiation or send offers for example. In such cases, these uncertainties are left mostly to the agent designers to model and use in their bargaining strategy (Faratin et al., 1998; Jennings et al., 2001) (i.e. in this case a strategy is a mapping from the history of offers to the next offer to be generated). To this end, the agents' owners may use some form of heuristic that provide general rules on how to add issues to the negotiation object, the type of offer to be sent, or the trade-offs that can be made between different issues. The way these different functions are performed define the agent's negotiation *stance* (i.e. how it shapes the negotiation encounter to its advantage). Heuristics generally try to reach good outcomes (i.e. those that give a high positive utility to the participating agents) rather than optimal ones (Jennings et al., 2001). In contrast, optimal outcomes that maximise the sum of the utility of participating agents are usually sought by game-theoretic techniques (Nash, 1953; Muthoo, 1999). In this context, optimal outcomes are those that maximise the sum of the utility of participating agents. To achieve this outcome, the agents' preferences and all their possible actions are usually assumed to be known. However, as can be seen, such approaches often make overly strong assumptions about the availability of information about the agents' private preferences and action set.

- Mechanism Design (MD) — this involves the development of a protocol specifying an exact sequence (and number) of actions (imposed by the system designer) to ensure that agents act in such a way that the resulting behaviour satisfies certain properties sought for by the system designer (Dash et al., 2003). To this end, the system designer assumes that the agents present in the system interact in a *game-theoretic* way (meaning that each agent models the effect of its actions on other agents' actions). The mechanism thus devised is to ensure that, at *equilibrium*, the intended properties are satisfied. The equilibrium here determines the state reached when all agents choose their utility maximising course of action and the main properties sought for by such mechanisms include: (i) pareto efficiency (i.e. maximising the sum of the utility of all agents in such a way that no other allocation exists where an agent gains more utility and no other agent is worse off); (ii) incentive compatibility (i.e. enforcing truthful revelation about the agents' preferences or other attributes); and (iii) individual rationality (i.e. agents are better off participating in the mechanism than opting out). To achieve such properties, game-theoretic mechanisms generally assume a completely known action set and that each agent knows its preferences perfectly (but not those of its opponent). To achieve such outcomes, the system designer provides incentives to agents to behave in a certain way through the specification of a payment scheme (i.e. how payments are made to agents which sell goods) and an allocation scheme (i.e. how goods are allocated to agents which pay for them) that takes into account the utility-maximising nature of agents. Usually, the protocols used in mechanism design imply a centralised authority that regiments the interactions (i.e. decides the agreements for the agents after knowing their preferences).

In general, both bargaining and mechanism design are subject to some uncertainty regarding similar or different attributes. For example, mechanism design reduces the uncertainty about the agents' preferences by enforcing a protocol which elicits these preferences. In contrast, bargaining seeks to elicit these preferences through an iterative exchange of offers which is not guaranteed to find an agreement that satisfies the agents' preferences. Therefore, as shown in figure 1.1, there exists a number of attributes that are subject to uncertainty and we view these as a cloud that envelops the negotiation process. Here we will concentrate on the attributes that most obviously affect negotiations such as:[3]

- Honesty — in competitive interactions agents may lie about their preferences or reliability in order to maximise their utility and this may, in turn, lead to inefficiency in the system. In such cases, the system designer needs to provide the right incentives to elicit truthful revelation of such information. This is usually achieved through engineering the protocol using some form of game theoretic analysis (i.e. mechanism design). In cases where this is not possible, agents may analyse the honesty of their opponents over multiple encounters and avoid those that are most dishonest in the long run.

- Reliability — in cases where a negotiation opponent's reliability of performing a particular task is not perfect, an agent might want to add some more stringent conditions to the agreement reached between them (e.g. specify a quality standard to be met or a compensation to be paid if expectations not met). This aims to make sure that the enactment of the agreement by the opponent is in line with what the agent expects. In such cases, in order to be able to analyse the reliability of an opponent, the agent may need to model this attribute statistically over multiple encounters and elicit a decision from that model at negotiation time. In this context, the reliability and honesty of agents is captured through the concept of *trust* (see chapter 3 for more details).

  Definition 1.5. Trust is a belief an agent has that the other party will *do what it says it will* (being honest and reliable) or *reciprocate* (being reciprocative for the common good of both), given an *opportunity to defect* to get higher payoffs (adapted from (Dasgupta, 1998)).

  Thus, through a trust model, it is possible to capture the probability of losing utility in an interaction with a particular agent by virtue of its trustworthiness (i.e. its reliability and honesty). Hence, through the use of a trust model, the risk[4] that agents incur in interactions can be significantly reduced.

- Preferences — when each agent in a negotiation encounter knows its opponents preferences, the outcome is usually easy to predict according to game theory (Mas-Colell et al., 1995). In mechanism design, the protocol is usually devised in such a

[3]Other attributes, such as the communication mechanism used or the computational capability of the agents, are also subject to uncertainty, but in this thesis we will assume these are already factored into the decision making models of the agents.

[4]We conceive of an environment as being prone to uncertainty, when every possible event in the environment has an equal chance of happening. Risk, instead, arises when there is a probability that an event causing some utility loss will happen (Zeckhauser and Viscusi, 1990). These probabilities can be hard to estimate especially in the types of open distributed systems in which we are interested.

way that these preferences are elicited. However, when preferences are not known and agents are in a bargaining encounter, they have to use efficient techniques to search the space of offers that meets their opponent's preferences. To assist in this process, the agents could also exchange more information (on top of an offer) which gives partial information about their preferences (i.e. without completely revealing them).

- Environment (action set) — when agents do not know each other's possible actions, it is hard to act strategically (as per game theory) to find an agreement (which dictates a set of actions to the participants) that maximises the utility of participating agents. Moreover, if the space of all possible actions is very large, negotiating agents may find it computationally hard to find a solution in a negotiation encounter. In such cases, the system designer might need to formulate a protocol that reduces the space of actions that agents need to search to find an agreement.

Against this background, in this thesis we aim to develop models that can reduce the impact of the above uncertainties on the effectiveness of bargaining and mechanism design techniques. In general, this can be achieved either by engineering new protocols or enriching the strategy of an agent in order to make the system, as a whole, more robust to uncertainty.

In more detail, in bargaining models in multi-agent systems, the uncertainty about preferences and the environment are increasingly being researched using a new class of techniques, here termed *argumentation-based negotiation* techniques, of which persuasive negotiation is a special category (see chapter 2). These models attempt, in various ways, an exploration of agents' preferences and actions. Currently, such models limit themselves to very abstract implementations (i.e. make no connection to a real application). Moreover, no existing agent-based bargaining model deals with the uncertainties underlying the reliability of agents or the honesty of agents. Similarly, in mechanism design where action sets are assumed to be known and honesty is elicited, some attention has been given to the uncertainty with respect to preferences of agents (Mas-Colell et al., 1995). However, there is a dearth of mechanisms that deal with uncertainty about the reliability of agents.

Given these lacunae, we aim to develop a new persuasive negotiation mechanism that aims to achieve better outcomes in less time than current bargaining techniques. To this end, we will clearly specify both the protocol and the strategies of the participating agents in such a way that the uncertainty about the agents' action sets and preferences is reduced. We also aim to develop modelling techniques, based on the concept of trust, that can be used by agents to reduce the uncertainty they have about their counterparts' reliability and honesty both in bargaining and mechanism design. In so doing, we will develop mechanisms that can generate better outcomes than current models when faced with uncertainty. Finally, we aim to show the applicability of our models by providing an example application where our persuasive negotiation mechanism and trust model can be used.

In the following sections we outline the landscape within which we develop our models. We will therefore describe issues that need to be dealt with in the area of trust and argumentation-based negotiation respectively.

1.3 Trust in Multi-Agent Systems

Broadly speaking, there are two main approaches to trust in multi-agent systems which we will focus on in this thesis. Firstly, to allow agents to trust each other, there is a need to endow them with the ability to reason about the reciprocative nature, reliability or honesty of their counterparts. This ability is captured through trust models. Such models aim to enable agents to calculate the amount of trust they can place in their interaction partners. A high degree of trust in an agent would mean it is likely to be chosen as an interaction partner and (possibly) a reciprocative strategy used towards it over multiple interactions in order to elicit the best pay-off in the long run (Axelrod, 1984). Conversely, a low degree of trust in an agent would result in it not being selected (if other, more trusted, interaction partners are available) or a non-reciprocative strategy adopted against it over *multiple interactions* (if there is no better alternative). In this way, trust models aim to guide an agent's decision making in deciding on how, when, and who to interact with. However, in order to achieve this, trust models initially require agents to gather some knowledge about their counterparts' characteristics. This can be achieved in a number of different ways including: (i) through inferences drawn from the outcomes of multiple direct interactions with these partners forming the agent's *confidence* in them or (ii) through indirect information provided by others forming the *reputation* of these partners. The combination of an agent's confidence and reputation measures (through some decision mechanism) can then be used to derive a general notion of trust that the agent has in its counterparts.

Secondly, while trust models pertain to the reasoning and information gathering ability of agents, the other main approach to trust concerns the design of protocols of interactions (i.e. through mechanism design techniques). As stated in section 1.2, one of the main aims of MD is to devise systems that are incentive compatible. This is normally achieved by providing the right incentives in the form of payments that are made from the mechanism to the agents involved in it. Thus, agents are compelled to be honest by the system.

From these two perspectives, it can be seen that trust pervades multi-agent interactions at all levels (i.e. at the protocol level and at the agent's reasoning level). With respect to designing agents and open multi-agent systems we therefore conceptualise trust in the following ways:

- **individual-level trust**, whereby an agent has some beliefs about the honesty, reliability, or reciprocative nature of its interaction partners.
- **system-level trust**, whereby the actors in the system are forced to be honest by the rules of encounter (i.e. protocols and mechanisms) that regulate the system.

The above approaches can be seen as being complementary to each other since they suit different contexts. Thus, while protocols aim to ensure the honesty of agents at the system level, they are limited in that they require a central authority (to compute outcomes or receive private information) and assume agents are completely reliable. In contrast, where the system cannot be completely centralised and agents cannot be assumed to be completely reliable, trust models at the individual level provide an alternative approach to measuring trust in a distributed fashion and are only limited by the agents' own sensing and reasoning capability (see chapter 3 for more details).

As can be seen from figure 1.2, while the individual level trust models enable an agent *to reason* about its level of trust in its opponents, the system level mechanisms aim to *ensure* that these *opponents' actions* can actually be trusted. In more detail, using their trust models, agents can:

- reason about strategies to be used towards trustworthy and untrustworthy interaction partners (e.g. being reciprocative or selfish towards them) given a calculation of payoffs over future interactions (i.e. using learning and evolutionary models).
- reason about the information gathered through various means (e.g. either directly or through reputation models) about potential interaction partners (i.e. using reputation models).
- reason about the motivations and capabilities of these interaction partners to decide whether to believe in their trustworthiness (i.e. using socio-cognitive models).

In contrast, the mechanisms and protocols described (i.e. enforcing system-level trust) aim to force agents to *act and interact* truthfully by:

- imposing conditions that would cause them to lose utility if they did not abide by them (i.e. using trustworthy interaction mechanisms).
- using their reputation to promote their future interactions with other agents in the community or demote future interactions whenever they do not behave well (i.e. using reputation mechansims).

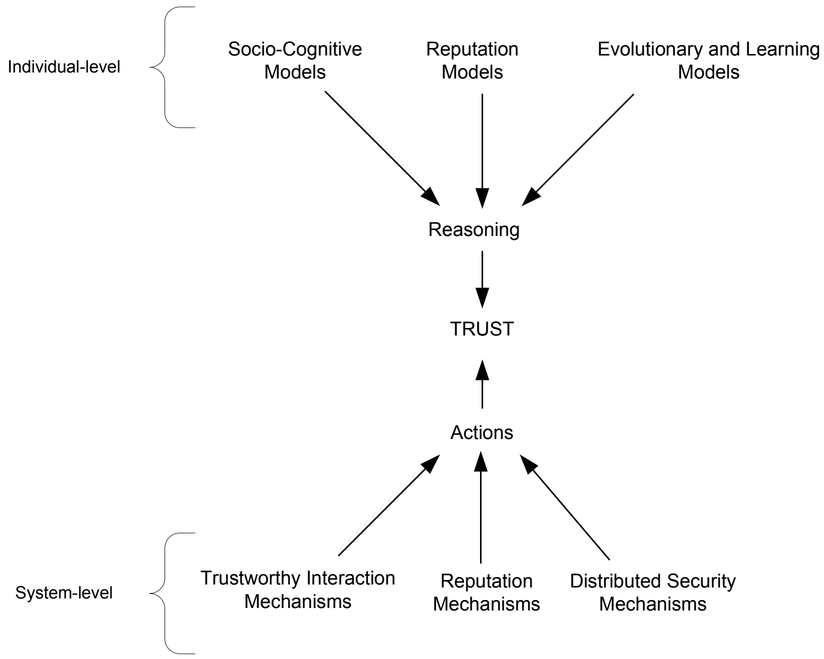


FIGURE 1.2: A classification of approaches to trust in multi-agent systems.

- imposing specified standards of good conduct that they need to satisfy and maintain in order to be allowed in the system (i.e. using security mechanisms).

In general, these two approaches to trust have, however, rarely been used to deal with the uncertainties that arise in negotiation (except in the process of partner selection, see chapter 3 for more details). For example, in bargaining no trust modelling technique has been devised to allow agents to influence agreements according to the believed reliability or honesty of their counterparts. In mechanism design, on the other hand, while most models have focused on incentive compatibility (honesty) as a means of eliciting trust, very few models deal with the varying reliability of agents (see chapter 6 for more details).

Given these observations, the first main aim of this thesis is to devise a trust model that captures both the reliability and honesty of agents, and can be used in both bargaining strategies and mechanism design. Obviously, the reliability or honesty of agents cannot easily be measured unless the agents observe each other's behaviour over a number of interactions or share information about their observations with other agents. A trust model therefore needs to build its measures of trust over multiple interactions to obtain a more precise impression of an opponent. Given this, an agent using such a trust model in a bargaining encounter would need to adjust its strategy over multiple encounters. Moreover, using a trust model in a mechanism would mean refining the computed outcome to choose those agents that are most reliable and honest in the long run.

The second main aim of this thesis is to deal with uncertainties concerning the preferences and actions of the agents. While mechanism design tends to assume known action sets and uses the protocol to elicit preferences in a mostly centralised fashion, bargaining approaches do not assume anything about action sets and do not specify a rigid protocol to elicit the true preferences of the agents. Rather, agents are left to use their own strategy in bargaining to find an agreement that maximises their utility. Bargaining is therefore very appropriate for distributed applications and is less restrictive than centralised mechanisms. However, bargaining in the simple form of exchange of offers and counter-offers does not allow an efficient exploration of preferences or action sets. In the next section we therefore explore a new bargaining approach that aims to make this exploration more efficient.

1.4 Argumentation-Based Negotiation

In section 1.3 we proposed to deal with uncertainties regarding the honesty and reliability of agents in bargaining encounters through the use of trust models. However, trust models do not cater for uncertainties about the preferences of the agents and the action sets. While in mechanism design these attributes are assumed to be available or are elicited through the protocol, in bargaining encounters these uncertainties are not usually taken into account and may lead to an inefficient outcome in the following ways:

- Uncertainty about preferences — the offers made during a bargaining encounter determine the values that one or multiple issues in the negotiation object must take in a possible agreement (e.g. price, quality, quantity). The domain of these values may be very large (because the agents may accept many different values for the same issues or trade-off the utility they obtain from the value of a particular issue for more utility for a value of another issue). During bargaining, the exchange of offers equates to searching through this large, and possibly multi-dimensional, space for an agreement. This may, therefore, prove to be a time-consuming and computationally expensive process. Also, the smaller the space of agreements that may satisfy all the negotiating agents' preferences, the harder it is to find the agreement in that large space of all offers. In such cases, agents are less likely to reach an agreement if they have short time deadlines or incur some costs in communicating their offers for example.

- Uncertainty about the action set — only involving a restricted number of issues in the negotiation object (e.g. price, quality, quantity) ignores the fact that the agents may have preferences about other resources or issues (upon which they can act) which may be negotiated. For example, a router agent might propose to give free access to some files to a client agent (which the client might be interested in and which do not cost anything to the router agent) if it accepts to pay a high price for bandwidth to access the internet. To ignore these issues in the negotiation process may reduce the possibility of finding common interests between agents which could, in turn, lead to an agreement that satisfies all the agents' preferences (Fisher and Ury, 1983). However, enlarging the space of issues needs to be undertaken carefully since this equates to an increase in uncertainty about the preferences of the agents over these issues.

The ideal negotiation procedure would therefore enable agents to quickly identify a small space of issues that they all value most. Given this, a new approach to bargaining has been growing in the past few years in the MAS community known as Argumentation-Based Negotiation (ABN). This approach involves the use of additional constructs in offers exchanged so as to make these offers more attractive to an opponent and therefore reach an agreement faster. These constructs aim to provide additional information about the agents' properties, resources, or attributes or about the offer made, that can reduce the uncertainty about their action set and preferences (without revealing their exact preferences). In so doing, this information reduces the time to find an agreement by allowing the agents to search a small number of issues they value most. ABN is based on constructs called *arguments* which we define as follows:

Definition 1.6. An argument is an illocution (a speech act (Searle, 1969; Austin, 1975)) that contains a justification for an offer or a commitment to some course of action conditional on whether the offer is accepted or not.

The above definition captures the two general strands of current research in ABN namely those that deal with justifications (or information) and those that deal with commitments to actions. These two types of arguments aim to achieve the similar objectives. In more detail, these two types of arguments make the search for an agreement between negotiating agents more efficient in the following ways:

- Justification-based ABN — justifications usually expose more information about an agent's preferences. These justifications either give more details about why an offer is rejected or what changes would be needed to make an offer acceptable (Fisher and Ury, 1983; Rahwan et al., 2003d). For example, if a seller offers a blue car for five hundred dollars, it may additionally argue that the car has a very good engine and that the blue colour is very trendy. These may, in turn, influence the buyer to increase its preference for that colour (since an agent's preferences may be partially determined by what other agents consider to be trendy). Moreover, such justifications can expose additional issues of common interest for the agents concerned (i.e. here the type of the engine or the colour).

- Persuasion-based ABN — commitments to actions conditional on the answer of the recipient (i.e. whether the agent accepts or rejects) determine the *persuasive tactics* that can be used in negotiation (Schelling, 1963). Such conditional commitments provide an agent with a means to influence its opponent through persuasive arguments such as threats (which are enacted if an offer is rejected) or rewards (which are enacted if an offer is accepted) in order to get an agreement faster. Their influence is captured by the constraints they impose on the domain of values an opponent may offer (or counter-offer) and on the space of issues that need to be considered in a given interaction. For example, a seller may promise to reward a buyer agent with a discount on its next purchase if it accepts to buy a car for six hundred pounds. Thus, the price of the next purchase is already biased in favour of the buyer and makes the search for an agreement faster (since values of the next negotiation object may be bounded by the reward). If a seller instead threatened to increase the price of future car services if its negotiation opponent did not accept a current offer about the price of the car being proposed, the opponent may be forced to accept the offer (if the threat is credible), since refusing may result in constraining the outcome of future negotiations about the car services to its detriment (and reducing the search for an agreement in the next encounter).

Both classes of arguments have their own merits but in this thesis we focus on persuasion based ABN, more commonly known as Persuasive Negotiation (PN), because we believe that using justifications is overly restrictive since this requires the *same* deductive mechanism for all negotiating agents. Also, each agent would need to believe what its opponent provides as a belief as completely true (i.e. they are trustworthy) (Amgoud et al., 2002; Parsons et al., 1998; Sadri et al., 2002). In comparison, PN provides a more established and less restrictive framework based on the use of commitments to actions which makes them more appealing in the following ways. First, PN does not impose any restriction on the deductive machinery used by negotiating agents. This means that the actions committed to in an argument can be evaluated through a preference structure such as a utility function. Second, PN does not assume that the agents are completely trustworthy and is therefore more applicable to competitive interactions where agents may lie about their preferences or reliability.

In general, a PN mechanism requires all of the following (Jennings et al., 2001):

1. Mechanisms must exist for passing proposals and their supporting arguments in a way that all agents involved understand them. This implies that the protocol needs to specify illocutions for agents to express the nature of the argument they need to send (e.g. whether they want to ask for a reward or give one).

2. Techniques must exist for generating proposals and for providing the supporting arguments. This implies that an agent must have a means to construct proposals given its own goals and the issues that are to be negotiated. It should also be able to devise a supporting argument about issues or actions that an opponent is expected to give value to.

3. Techniques must exist for assessing proposals and their associated supporting arguments. This usually involves evaluating the proposals and arguments according to the agent's preferences (i.e. utility function).

4. Techniques must exist for responding to proposals and their associated supporting arguments. This implies that an agent must have a strategy to generate offers and arguments. In particular, this involves measuring the utility that is likely to be obtained from a given combination of a received offer and argument and to give a best response that seeks to meet the opponent's preferences while still maximising the agent's utility.

In most existing PN mechanisms, however, arguments such as threats or rewards are usually given very abstract definitions that cannot readily be used in real applications (see chapter 2 for more details). Specifically, rewards or threats are generally defined as those actions that aim to increase or reduce the utility of an opponent, respectively, at a later point in time (after a negotiation encounter is terminated and an agreement is enacted or not). In such cases, those actions that are deemed credible rewards or threats represent yet another space the agents need to search in order to find the most appropriate one to send. To reduce the size of this space, in this thesis we decided to impose a protocol or a strategy that constrains the set of actions that can be considered rewards and threats. It therefore follows that the protocol and the rewards (or threats) need to be defined in such a way that they have clear semantics to be used in a realistic application.

Against this background, we believe PN is a natural fit for repeated negotiation encounters (or long-term relationships) for a number of reasons. First, arguments that apply to actions at a later point in time can easily be construed as constraints that apply over negotiations in future encounters (hence giving clear semantics to arguments). In this way, arguments can reduce the space of actions that needs to be searched for an agreement in future encounters (thus reducing uncertainty about the action set) and influence an agent to accept an offer in a present encounter (thus reducing the time to find an agreement that matches the agents' preferences). Second, by constraining future encounters through arguments, agents can ensure that they obtain a positive utility in the present encounter or future ones whenever they have an amount of resources or capabilities that can predictably vary over time. This can be achieved by applying constraints to future encounters that guarantee certain outcomes that fit the agents' dynamic constraints.

By applying PN to repeated encounters we therefore define the common thread that links this mechanism with our use of trust models in negotiation to reduce uncertainty about the reliability and honesty of agents. In effect, these models of PN and trust aim to provide an agent with a reasoning mechanism robust to the uncertainties which are endemic to repeated negotiations in open distributed systems. Moreover, we aim to show through example applications how they would each work in practice.

1.5 Research Contributions

In this section we summarize the aims and objectives of this thesis and the contributions to the state of the art that were made to achieve them. Our general aim is to develop techniques that help to reduce the uncertainty in repeated multi-agent negotiations in open distributed systems. To this end, we set out to achieve the following particular objectives:

- Develop a comprehensive trust model for multi-agent systems that can evaluate the reliability or honesty of agents.
- Use trust in bargaining encounters and mechanism design in order to reduce the uncertainty agents have about their negotiation opponents' reliability and honesty. This involves using the trust measure developed by our trust model in an agent's reasoning mechanism (i.e. in its bargaining strategy) and developing the protocol for an interaction mechanism that caters for the uncertainty regarding the reliability of agents.
- Develop a comprehensive model of persuasive negotiation that comprises: (i) a protocol that incorporates the use of arguments and determines what commitments hold whenever agents make offers or issue arguments and (ii) a reasoning mechanism that can generate offers and arguments and can evaluate and respond to these during a negotiation encounter. This requires developing both an argument generation and evaluation component, as well as the *strategies* for PN.
- Demonstrate the benefit of using arguments in automated negotiation and show that they enable agents to reach agreements more efficiently (i.e. better and faster) than using normal negotiation protocols that only allow an exchange of offers.
- Implement PN in a realistic context in order to demonstrate its applicability and effectiveness in managing inter-agent dependencies.

To achieve these objectives, a number of contributions were made to the state of the art:

- In (Rahwan et al., 2003b), we provided the first survey of the state of the art in the area of ABN and identified the main trends and challenges that pervade the field. This survey set the landscape within which we develop our model of PN and appears as chapter 2 in this thesis.
- In (Ramchurn et al., 2004b), we provided a critical analysis of the trust issues that arise in MAS. In particular, we showed how various models developed in MAS form a coherent approach towards resolving uncertainties about the reliability and honesty of agents. Thus, we also identify the current challenges in the field which we aim to meet in our model. This review appears in chapter 3 in this thesis.

- Based on our preliminary work in (Ramchurn et al., 2004d) and the requirements presented in chapter 3, we describe a novel trust model (called CREDIT) that enables an agent to measure its opponents' trustworthiness (honesty or reliability) over multiple encounters (Ramchurn et al., 2004c). The model is shown to be effective and efficient at preventing exploitation by opponents by allowing the agent to adjust its negotiation stance in repeated bargaining encounters according to its trust in its opponents (hence reducing uncertainty). Moreover, using CREDIT's trust measure, an agent is also able to select its interaction partners more effectively. CREDIT is presented in chapter 5.

- Given our work on CREDIT, we then introduced the use of trust modelling to the area of mechanism design by developing the notion of Trust-Based Mechanism Design (TBMD) to reduce uncertainty about the reliability of agents (Dash et al., 2003). In so doing, we created the first efficient, individually rational, and incentive compatible mechanism that takes into account the trust agents have in each other. This is, in effect, the first efficient reputation mechanism that incentivises agents to reveal their impressions of others truthfully. Specifically, our Trust-Based Mechanism combines these measures into an overall trust measure (using a trust model such as CREDIT) to select those agents that are best at doing certain tasks. This work is presented in chapter 6.

- While CREDIT and TBMD are concerned with uncertainties about reliability and honesty, in (Ramchurn et al., 2003a) we provided a preliminary model of Persuasive Negotiation whereby agents can use threats and rewards to elicit better agreements by reducing uncertainties about preferences and action sets. This model describes the general concepts that are used to develop our new PN mechanism that can allows agents to reach better agreements faster than standard bargaining mechanisms. In particular, we develop a new protocol and reasoning mechanism for agents to use to give or ask for rewards in repeated encounters. We also show, empirically, that agents are able to engage in more efficient and effective agreements using this protocol and reasoning mechanism than only bargaining with offers. We also develop a novel strategy for PN and show that it enables agents to achieve even better agreements than current negotiation strategies. The complete model is given in chapter 7.

- Given our model of PN, we then apply it, together with CREDIT, in a pervasive computing environment (Ramchurn et al., 2004a). In so doing, we are able to show, for the first time, how a PN and trust model can be used in practice to allow agents to resolve their conflicts effectively. In particular, in this work we show how PN can be used by agents to negotiate about the usefulness of interruptions in a meeting room scenario. In so doing, negotiating agents provide an effective way to reduce the intrusions caused by interruptions and help their human owners to focus on the main task undertaken during the meeting. This work appears as chapter 8.

Drawing all these together, the application of the various models we develop to cater for uncertainties in negotiation is graphically expressed in figure 1.3. As can be seen, CREDIT and TBMD overlap in that they deal with the uncertainty regarding the reliability and honesty of agents. In CREDIT, we show how to develop and use trust in bargaining encounters, while in TBMD we show how to use the core concepts of CREDIT in mechanism design. Given this, CREDIT also overlaps with persuasive negotiation as they both apply to bargaining encounters and both try to reduce uncertainty about the action set of agents. They do this by either adjusting the negotiation stance of an agent (i.e. the selection of values for issues in this context) or by using arguments to constrain the action set. In addition to this, PN overlaps with TBMD since PN also aims to explore preferences more efficiently than standard bargaining techniques through the use of arguments while TBMD aims to elicit these preferences through the protocol it enforces upon the agents together with the trust model it uses.

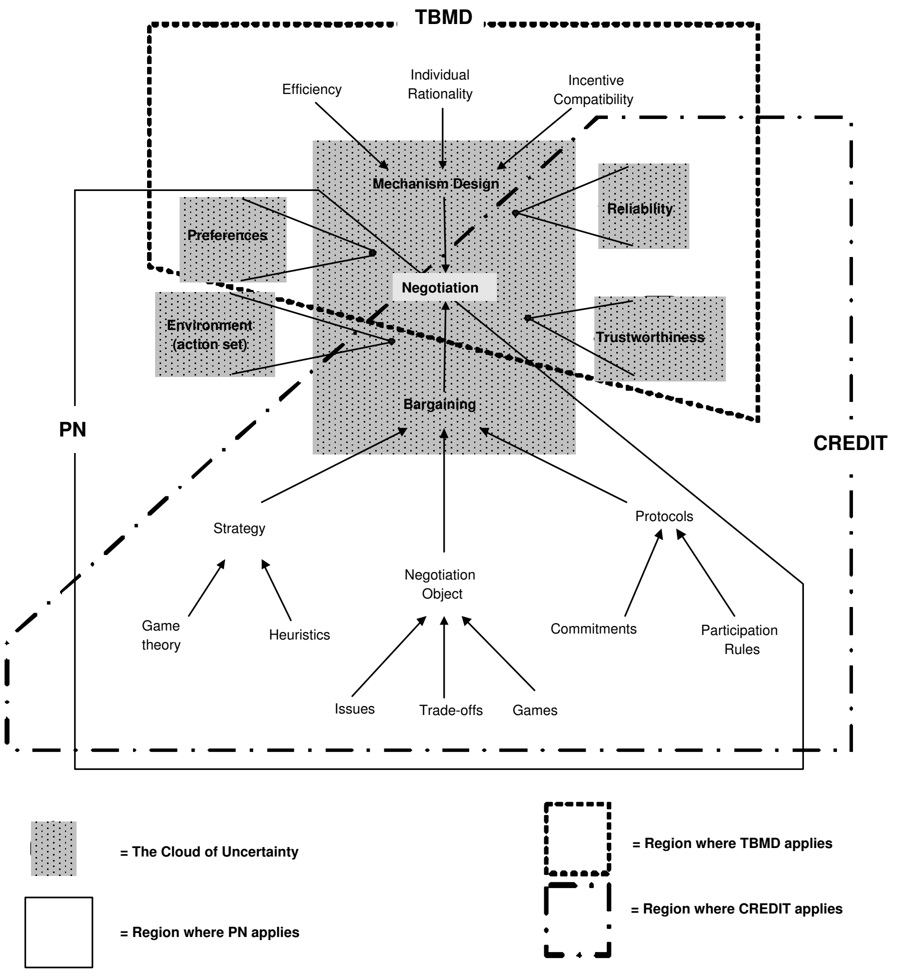


FIGURE 1.3: Applying CREDIT, TBMD, and PN to reduce the uncertainty underlying negotiation.

1.6 Thesis Structure

The thesis is structured in the following way:

Chapter 2 : surveys the literature on ABN. The main models in the literature are analysed and their prominent characteristics defined. We identify the particular components of an ABN protocol and strategy and discuss the main challenges that still exist in this area. We then define the requirements of the protocol and reasoning mechanism that are to be developed for our persuasive negotiation model which we describe in chapter 7.

Chapter 3 : details the background on trust for MAS. Here we justify the need for trust to be acquired through an agents' modelling capabilities and through constraints imposed by the interaction mechanism. Thus, different types of trust mechanisms are surveyed and a general typology of trust is derived. Following this, we define the main requirements for the CREDIT trust model and TBMD which we develop in chapters 5 and 6 respectively.

Chapter 4 : formalises the basic notions of agreements and preferences. These formalisms define the common thread that underpins the settings used in the empirical evaluation of CREDIT and PN.

Chapter 5 : details the CREDIT trust model. The model is based on the notions of confidence and reputation and can be used to influence an agent's negotiation stance. The computational complexity of the algorithm underlying the model is shown to be linear in the number of negotiated issues and quadratic with respect to the number of decision variables (here these are fuzzy sets) used to capture the particular behaviour of opponents. The model is shown to be effective and efficient at enabling an agent to avoid interacting with agents that are unreliable or dishonest. Moreover, CREDIT allows an agent to adapt its negotiation stance so as to elicit profitable outcomes when it deals with those agents that are reliable to some degree.

Chapter 6 : applies the basic notions of trust from CREDIT to design a centralised mechanism using game theoretic principles in order to elicit efficiency in the system. Thus, in this chapter, we develop TBMD as a novel method to deal with the variable reliability of agents. Specifically, we prove that the trust-based mechanism we develop is incentive compatible, individually rational, and efficient. Then we empirically show that our mechanism enables the most reliable agents to be chosen in the long run and that it performs better than other comparable models.

Chapter 7 : presents our novel model of PN based on the exchange of rewards (asked for or given). This involves detailing a protocol that guides the use of arguments in negotiation and manages the commitments that agents make through their offers and arguments. In particular, we use dynamic logic to build such a framework and we provide the reasoning mechanism that agents can use to generate offers and arguments (i.e. we define strategies for PN). Then, we empirically evaluate the model and show that it enables agents to achieve better agreements (i.e. higher utility) faster (i.e. in less negotiation rounds) than current bargaining protocols and strategies.

Chapter 8 : describes an application of our PN and trust models. In particular, through PN and CREDIT, we resolve the conflicting preferences agents have about the interruptions received on devices in a pervasive computing environment. The particular scenario chosen is that of a meeting room which contains devices (built into the room and those that each participant owns) on which notifications are received. In such a context, we therefore show how PN can be used to flexibly manage these notifications so as to minimise their intrusiveness while the meeting takes place.

Chapter 9 : summarizes and discusses the main achievements of this thesis. We analyse the extent to which the models therein meet the objectives set in chapter 1. Finally, we discuss future avenues of work that we have identified in the domains of trust and PN.

Chapter 2

Argumentation-Based Approaches to Negotiation

In chapter 1 we justified the need to build an ABN mechanism in general, and a PN mechanism in particular, to reduce the uncertainty that agents have about each other's preferences and action sets. Before doing so, however, in this chapter we first survey the state of the art in the area of ABN to determine the main issues that arise in designing such mechanisms. In particular, we analyse the impact of augmenting bargaining mechanisms with arguments. We do this both from the perspective of the system (comprising the negotiation object and the protocol) within which agents interact and the agent's decision making model. In so doing, we also define the main requirements of our PN mechanism and identify those elements of existing ABN mechanisms we can exploit in developing our model.

The rest of this chapter is structured as follows. In section 2.1 we discus the main components of the framework that is needed to implement an ABN mechanism. Given this, in section 2.2 we elaborate on the general requirements of ABN agents. Finally, section 2.3 summarises our findings and outlines the main challenges that exist in defining our PN mechanism.

2.1 External Elements of ABN Frameworks

As described in figure 1.1, in order to allow agents to resolve their conflicts, bargaining mechanisms require different elements in the agent's reasoning mechanism (i.e. its strategy) and in the system within which these agents interact (i.e. the protocol). Moreover, bargaining mechanisms require that agents are also able to understand the different operations that are possible on the negotiation object and are able to communicate these operations to their counterparts. To this end, agents also need a language to understand

and communicate their different manipulations of the object. In this context, therefore, the definition of the language and the protocol constitute the *framework* of what we consider to be a bargaining mechanism.

We will deal with approaches to the agents' decision making models (capturing their strategy and assessment of the negotiation object) in section 2.2 while in this section we will focus on the protocol and the language that allows an agent to manipulate the negotiation object. Considering the language first, this primarily involves defining different aspects of the *domain* that are necessary to describe the content of the negotiation object (e.g. what are the issues involved in the object or what are the agents that are to benefit from an agreement). Moreover, the language may also need to define those elements that allow agents to *communicate* their intended actions on the negotiation object (e.g. whether they are retracting their offer or requesting more information about the object). Turning now to the protocol, this can generally be divided into two main parts (as shown in figure 1.1); namely managing participation rules and managing the commitments agents make and the information they exchange during bargaining. In this context, the participation rules determine when and what particular actions can be carried out. This includes actions such as making an offer or withdrawing from the negotiation. The management of commitments requires that the system keeps track of what commitments agents make and what commitments are fulfilled (as a result of what they say or do). The management of these commitments or any information that is passed between the agents is usually captured by an information storage component. Given these different aspects of the system, in the following subsections we detail the different approaches to defining the language for bargaining, the participation rules, and the information stores.

2.1.1 The Language for Bargaining

A negotiation framework requires a language that facilitates communication between the agents (Labrou et al., 1999). Elements of the *communication language* are usually referred to as *locutions* or *utterances* or *speech acts* (Searle, 1969; Traum, 1999). Traditional automated negotiation mechanisms normally include the basic locutions such as `propose` for making proposals, `accept` for accepting proposals, and `reject` for rejecting proposals.

In addition to the communication language, agents often need a common *domain language* for referring to concepts of the environment, the different agents, time, proposals, and so on.[1] When a statement in the domain language is exchanged between agents, it is given particular meaning by the communication language utterance that encap-

[1] Note that this language may be different from the language used internally by an agent. In such cases, the agent needs to perform some type of *translation* into the common language in order for communication to work (Sierra et al., 1998).

sulates it. For example, in the PN framework presented by (Sierra et al., 1998), the locution `offer`$(a, b, \mathit{Price} = \$200 \wedge \mathit{Item} = \mathit{palm130}, t_1)$, means that agent a proposes to agent b, at time t_1 the sale of item *palm130* for the price of \$200. On the other hand, the `reject` locution gives the same content a different meaning. The locution `reject`$(b, a, \mathit{Price} = \$200 \wedge \mathit{Item} = \mathit{palm130}, t_2)$ means that agent b rejects such a proposal made by agent a.

In ABN frameworks, agents need richer communication and domain languages to be able to exchange meta-level information (i.e., information other than that describing outcomes). Therefore, a major distinguishing factor of ABN frameworks is in the type of information that can be expressed and exchanged between agents, and consequently, in the specifications of the agents that generate and evaluate this information. As we pointed out in section 1.4 chapter 1, such information are intended to allow a better reduction of uncertainties that surround agents' action sets and preferences. Table 2.1 shows the main distinguishing features between ABN and non-ABN frameworks as they relate to the communication and domain languages.

	Non-ABN Frameworks	**ABN Frameworks**
Domain Language	Expresses proposals only (e.g., by describing products available for sale).	Expresses proposals as well as meta-information about the world, agent's beliefs, preferences, goals, etc.
Communication Language	Locutions allow agents to pass calls for bids, proposals, acceptance and rejection, etc.	In addition, locutions allow agents to pass meta-information either separately or in conjunction with other locutions.

Table 2.1: Differences between ABN and Non-ABN w.r.t Domain and Communication Languages

2.1.1.1 State of the Art

In existing ABN frameworks, various domain and communication languages have been proposed. They range from those designed as simplistic domain specific languages to more complex languages grounded in rich logical models of agency.

In multi-agent systems, two major proposals for agent communication languages have been advanced, namely the Knowledge Query and Manipulation Language (KQML) (Mayfield et al., 1996) and the Foundation for Intelligent Physical Agents' Agent Communication Language (FIPA ACL) (FIPA, 2001). FIPA ACL, for example, offers 22 locutions. The contents of the messages can be in any domain language. The locution `inform`$(a, b, \varphi, \mathit{lan})$, for example, allows agent a to inform another agent b of statement

φ which is in language *lan*. Other locutions exist allowing agents to express proposals for action, acceptance and rejection of proposals, make various queries about time and place, and so on. FIPA ACL has been given semantics in the form of pre- and post-conditions of each locution (Searle, 1969; Austin, 1975).

While FIPA ACL offers the benefits of being a more or less standard agent communication language, it fails to capture all utterances needed in a negotiation interaction. For example, FIPA ACL does not have locutions expressing the desire to enter or leave a negotiation interaction, to provide an explicit critique to a proposal or to request an argument for a claim. While such locutions may be constructed by injecting particular domain language statements within locutions similar to those of FIPA ACL, the semantics of these statements fall outside the boundaries of the communication language. Consider the following locution from the framework presented by (Kraus et al., 1998):

$$\texttt{Request}(j, i, Do(i, \alpha), Do(i, \alpha) \rightarrow Do(j, \beta))$$

In this locution, agent j requests that agent i performs action α and supports that request with an argument stating that if i accepts, j will perform action β in return. For the locution to properly express a promise, action β must actually be desired by agent i. If, on the contrary, β is undesirable to i, the same locution becomes a threat, and might deter i from executing α. The locution `Request`, however, does not include information that conveys this distinction.

In order to deal with the above problem, ABN framework designers often choose to provide their own negotiation-specific locutions which hold, within them, the appropriate semantics of the message. For example, (Sierra et al., 1998) provides explicit locutions for expressing threats and rewards (e.g., $\texttt{threaten}(i, j, \alpha, \beta)$ and $\texttt{promise}(i, j, \alpha, \beta)$).

Having discussed some issues relating to the communication languages in ABN, let us now discuss the domain languages. In negotiation, the domain language must, at least, be capable of expressing the object of negotiation. In Sierra et al.'s model (Sierra et al., 1998), the domain language can express variables representing negotiation issues (or attributes), constants representing values for the negotiation issues (including a special constant '?' denoting the absence of value), as well as equality and conjunction. This enables them to express full or partial multiple-attribute proposals. For example the sentence

$$(Price = \pounds 10) \wedge (Quality = high) \wedge (Penalty = ?)$$

expresses a proposal to agree on a high-quality product or service for the price of $\pounds 10$, and with a cancellation penalty yet to be agreed upon. There is also a meta-language for explicitly expressing preferences. For example, the statement $Pref([Price = \pounds 10], [Price = \pounds 20])$ expresses the fact that an agent prefers a price of $\pounds 10$ to $\pounds 20$.

In addition, ABN frameworks may need some way to express plans and resources needed for different plans. This is because agents participating in negotiation may be doing so in order to obtain resources needed for executing their plans. This means that an agent may be able to inform another agent of (parts of) its plans in order to justify its request for particular resources. (Sadri et al., 2002), for example, express plans using the *plan*(.) predicate. The following formula:

$$plan(\langle hit(nail), hang(picture)\rangle, \{picture, nail, hammer\})$$

denotes a plan (or intention) to hit a nail and hang a picture. The resources this plan requires are a picture, a nail and a hammer.

Some ABN frameworks also explicitly express information about agents' mental attitudes. The ABN frameworks presented by (Kraus et al., 1998) and by (Parsons et al., 1998), for example, allow an agent to represent beliefs about other agents' beliefs, desires, intentions, capabilities, and so on, and are based on logics of Belief, Desire, and Intention (BDI) (Rao and Georgeff, 1995; Wooldridge, 2002). An agent can use this information not only in its internal reasoning processes, but also in its interaction with other agents.

The usefulness of the domain language in the context of ABN becomes particularly apparent when agents provide arguments for requesting certain resources, for rejecting certain requests, and so on. The richer the domain language, the richer the arguments that can be exchanged between agents. This will become more evident when we discuss argument generation and evaluation in the following sections.

2.1.1.2 Challenges

There are a number of challenges in the design of domain and communication languages for ABN. First, there is a need to provide rich communication languages with clear semantics. For example, McBurney et al. (2003) specified a set of locutions as part of a dialogue game[2] for purchase negotiation among multiple agents. The authors provided a public axiomatic semantics to their locutions by stating each locution's externally observable preconditions, the possible response, and the updates to the information stores.[3] Moreover, the framework presents an operational semantics of the whole framework, connecting locutions with each other via the agents' decision mechanisms. However, this framework does not cover the whole spectrum of ABN situations. For example,

[2]Dialogue games are interactions between two or more players, where each player makes a *move* by making some utterance in a common communication language, and according to some pre-defined rules. Dialogue games have their roots in the philosophy of argumentation (Aristotle, 1928; Hamblin, 1970). In multi-agent systems, dialogue games have been used to specify dialogue protocols for persuasion (Amgoud et al., 2000a), negotiation (Amgoud and Parsons, 2001), and team formation (Dignum et al., 2000).

[3]We shall discuss information stores in more detail in section 2.1.3.

there are no locutions for explicitly requesting, providing and challenging arguments, or for supporting argumentation over preference criteria. Locutions facilitating argument exchange have been proposed in other frameworks (e.g., Sadri et al., 2001a; Torroni and Toni, 2001; Sadri et al., 2002; Amgoud et al., 2000b; Amgoud and Parsons, 2001). There are opportunities for extending the model of (McBurney et al., 2003) with a richer argumentation system. Moreover, there is also a need to extend the semantics of the language to cover other actions (e.g. the enactment of an agreement or a reward) apart from illocutions. This may require the use of an action-based logic (as opposed to first order logic which is commonly used) such as dynamic logic (Harel, 1984).

2.1.2 Participation Rules

As discussed in chapter 1, a bargaining framework should also specify a *participation rules* in order to constrain the use of the language and other the actions of the participants (i.e. what the agents can say or do at what point in time). Thus, participation rules reduce uncertainty about the actions of the agents. Here we view the participation rules as a formal set of conventions governing the interaction among participants (Jennings et al., 2001). This includes the *dialogue rules* as well as other rules governing the actions of the agents.

The dialogue rules specify, at each stage of the negotiation process, who is allowed to say what. For example, after one agent makes a proposal, the other agent may be able to accept it, reject it or criticise it, but might not be allowed to ignore it by making a counterproposal. The rules might be based solely on the last utterance made, or might depend on a more complex history of messages between agents.

The other rules that form part of the participation rules may address the following issues (Jennings et al., 2001; Esteva et al., 2001):

- **Rules for admission**: specify when an agent can participate in a negotiation dialogue and under what conditions.
- **Rules for participant withdrawal**: specify when a participant may withdraw from the negotiation.
- **Termination rules**: specify when an encounter must end (e.g. if one agent utters an acceptance locution).
- **Rules for proposal validity**: specify when a proposal is compliant with some conditions (e.g., an agent may not be allowed to make a proposal that has already been rejected).
- **Rules for outcome determination**: specify the outcome of the interaction. In an auction-based framework, this would involve determining the winning bid(s)

(Sandholm, 2002). In argumentation-based frameworks, these rules might enforce some outcome based on the underlying theory of argumentation (e.g., if an agent cannot construct an argument against a request, it accepts it (Parsons et al., 1998)).

- **Commitment rules**: specify how agents' commitments should be managed, whether and when an agent can withdraw a commitment made previously in the dialogue, how inconsistencies between an utterance and a previous commitment are accounted for, and so on. These rules make the connection between information stores (which we discuss in the next section) and the dialogue rules.

In ABN, the participation rules are usually more complex than those in non-ABN. By "more complex", we mean that the participation rules may have to consider a larger number of locutions, and, hence consist of a larger number of rules. This leads to computational complexity arising from processes such as checking the locutions for conformance with the protocol given the history of locutions.

2.1.2.1 State of the Art

With respect to the dialogue rules, a variety of trends can be found in the ABN literature. Dialogue rules can be either specified in an *explicit* accessible format, or be only *implicit* and hardwired into the agents' specification.

Explicit specification of dialogue rules may be represented by finite state machines (e.g., Sierra et al., 1998; Parsons et al., 1998). In this way, each state has a set of outgoing and incoming transition lines that represent illocutions that bring the agents and take the agents away from that state. Thus, a *propose* illocution in (Sierra et al., 1998) brings an agent from one state where the only other possible illocution is *withdraw* to one where it is additionally possible to say *accept* (and *propose*). Another representation of this state machine can be achieved by expressing dialogue rules explicitly as in dialogue games (e.g., as in Amgoud et al., 2000b; Amgoud and Parsons, 2001; McBurney et al., 2003) by stating the pre- and post-conditions of each locution as well as its effects on agents' commitments. The following is the specification of a locution from the protocol presented by (McBurney et al., 2003). This locution allows a seller (or advisor) agent to announce that it (or another seller) is willing to sell a particular option:[4]

Locution: $\texttt{willing\_to\_sell}(P_1, T, P_2, V)$, where P_1 is either an advisor or a seller, T is the set of participants, P_2 is a seller and V is a set of sales options.

[4]We leave the discussion of "information stores" to section 2.1.3.

Preconditions: Some participant P_3 must have previously uttered a locution `seek_info`(P_3, S, p) where $P_1 \in S$ (the set of sellers), and the options in V satisfy constraint p.

Meaning: The speaker P_1 indicates to audience T that agent P_2 is willing to supply the finite set $V = \{\bar{a}, \bar{b}, \ldots\}$ of purchase-options to any buyer in set T. Each of these options satisfy constraint p uttered as part of the prior `seek_info`(.) locution.

Response: None required.

Information Store Updates: For each $\bar{a} \in V$, the 3-tuple $(T, P_2, \bar{a})$ is inserted into $IS(P_1)$, the information store for agent P_1.

Commitment Store Updates: No effects.

One advantage of dialogue game protocols is that they have public axiomatic semantics. This is because they refer only to observable pre-conditions and effects, rather than to the agents' internal mental attitudes. This makes it easier to verify whether agents are conforming to the protocol.

Other frameworks implicitly hardwire the dialogue rules in the agents' internal specification (e.g., Kraus et al., 1998; Sadri et al., 2001a; Torroni and Toni, 2001; Sadri et al., 2001b, 2002). In these frameworks, the dialogue rules are specified using logical constraints expressed in the form of if-then rules. Since these frameworks describe a logic-based approach to agent specification ((Kraus et al., 1998) implement their agents using Logic Programs, while (Sadri et al., 2001b) use Abductive Logic Programs), the participation rules are coded as part of the agent's program. These rules take the form $P(t) \wedge C(t) \Rightarrow P'(t+1)$, meaning that if the agent received performative (i.e. locution) P at time t, and condition C was satisfied at that time, then the agent *must* use the performative P' at the next time point. The condition C describes the rationality precondition in the agent's mental state. For example, one rule might state that if an agent received a performative which includes a request for a resource and it does not have that resource, then it must refuse the request. Note that this constitutes a private semantics of the protocol, and is hence harder to enforce by an external regulator (which might be needed to ensure the predictability of the system).

The termination rules in negotiation protocols specified as finite state machines are defined as a set of links to a final state. This is usually the case when one agent utters a `withdraw`(.) or an `accept`(.) locution. In the framework of (McBurney et al., 2003), a rule specifies that the dialogue ends after an agent utters the locution `withdraw_dialogue`(.) causing either no remaining sellers or no remaining buyers in the dialogue. In some frameworks, however, no termination rules have been defined, and hence the dialogue remains open even after agreement or failure.

In relation to outcome determination rules, some frameworks determine outcomes based on the logical structure of interacting arguments. For example, in the frameworks of

(Parsons et al., 1998) and (Amgoud et al., 2000b), a rule specifies that an agent must accept a request if it fails to produce an argument against that request. A similar case occurs when agents argue about their beliefs — an agent must accept a proposition if it fails to provide an argument for the negation of the proposition. In this sense, outcome determination is *implicit* in the underlying argumentation logic. In other frameworks, such as those of (Kraus et al., 1998), outcomes are reached through uttering a specific locution *explicitly* (e.g., by uttering `accept(.)`). Agents may utter such a locution based on some internal utility evaluation.

We shall leave the discussion of commitment rules to section 2.1.3, where we discuss information stores.

2.1.2.2 Challenges

Participation rules for ABN share the challenges faced in the design of argumentation protocols in general. For example, there is a need for qualities such as fairness, clarity of the underlying argumentation theory, discouragement of disruption by participants, rule consistency, and so on.[5]

One particularly important property is that of *termination*. To this end, some rules for preventing certain causes of infinite dialogues have been proposed. For example, the protocol of (Amgoud and Parsons, 2001) does not allow agents to repeat the exact same locutions over and over again. The intuition is that this would prevent the agent from, say, repeating the same question over and over again. In subsequent papers, the authors present further analysis of the outcomes of various argumentation-based dialogues (Parsons et al., 2002, 2003).

Torroni (2002) study termination and success in an ABN framework presented earlier (see Sadri et al., 2001b). Since the ABN framework is grounded in an operationally defined agent architecture based on abductive logic programming, it has been possible to study some properties by referring to the machinery of abduction. In particular, the author determined an upper limit to the maximum length of a dialogue, measured in the number of exchanged messages. Since these results are strongly dependent on the underlying logical system, it is not clear whether these results can be generalised to a variety of protocols without regard to the internal agent architecture.

Another important desired property in ABN paricipation rules is that of *guaranteed success*. Wooldridge and Parsons (2000) investigate the conditions under which particular logic-based negotiation protocols terminate with agreement. They provide results showing the complexity of solving this problem with negotiation frameworks using different domain languages. Most interestingly, they show that the problem of determining

[5]For a more elaborate discussion of the properties desired in argumentation protocols, refer to (McBurney et al., 2002).

whether a given protocol can be guaranteed to succeed, when used with a FIPA-like communication language, is provably intractable.

An important problem related to participation rules in general is that of *conformance checking*. This problem is concerned with answering the question of whether a particular utterance is acceptable, given the history and context of interaction. Conformance checking is one of the sources of complexity in dialogue systems; however, to date, it has received little attention in the ABN literature.

Another avenue of future research is in the design of admission rules in negotiation protocols. While some frameworks (e.g., McBurney et al., 2003) require that agents explicitly request to enter a negotiation dialogue no ABN framework includes external rules that govern admission to the negotiation dialogue. One may envisage situations where only agents with particular credentials, such as reputation or performance history, may be admitted to a negotiation. For example, a malicious agent may attempt to disrupt the interaction among other participants, and hence should not be admitted. Moreover, in repeated encounters, agents usually continue to interact only if they come to a fruitful outcome in the previous interactions (e.g. in making short term contracts for a long-term project or buyers choosing those sellers repeatedly only if their services prove to be of a good quality each time). More work needs to be done on investigating the effect of different admission rules on the outcome of negotiation.

2.1.3 Information Stores

In some ABN frameworks, there is no explicit centralised information store available. Instead, agents internally keep track of past utterances (e.g., Kraus et al., 1998). However, in many negotiation frameworks there is a need to keep externally accessible information during interaction. For example, we might need to store the history of utterances for future reference or to store information about the reputation of participants (Yu and Singh, 2002a; Rubiera et al., 2001). Moreover, having external information stores makes it possible to perform some kind of *enforcement* of protocol-related behaviours. For example, we may be able to prevent an agent from denying a promise it has previously made.

2.1.3.1 State of the Art

One type of information store that is common in the argumentation literature is the *commitment store*.[6] Commitment stores were initially conceived by (Hamblin, 1970) as a way of tracking the claims made by participants in dialogue games. Hamblin studied

[6]For a more detailed discussion of commitments in multi-agent dialogues, see (Maudet and Chaib-draa, 2003).

dialogues over beliefs, although he was at pains to state that commitments made in dialogue games should not be construed as necessarily representing the real beliefs of the respective participants (Hamblin, 1970, p. 257). Hamblin's notion of commitment store has been influential in later work on dialogue games, both in philosophy and in MAS, although the notions of commitment used sometimes differ. In the work on the philosophy of dialogue (e.g., Walton and Krabbe, 1995) the focus is on action commitments, i.e., promises to initiate, execute or maintain an action or course of actions. Commitments to defend a claim if questioned, called propositional commitments, are viewed as special cases of such action commitments by these authors. In the MAS literature the concern is usually with action commitments, where the associated actions are assumed to take place outside the agent dialogue. For example, one agent may commit to providing a specified product or service to another agent.

Note that commitment stores should not be confused with the interaction history, which only records the sequence of utterances during the whole interaction.[7] While the latter only form a passive storage of "unprocessed" utterances, commitments in commitment stores have more elaborate consequences. For example, when an agent asserts a proposition p, it may not only be committed to believe that p holds, but also to defending that p (if challenged), not denying that p, giving evidence that p, and so on (Walton and Krabbe, 1995). In the MAS literature, (Singh, 2000) gave "social" semantics for commitments using modal operators in branching-time logic. This semantics is public (i.e., is based on external observations of utterances as opposed to agents' internal mental states) and hence can be used for specifying, and checking for conformance with, the interaction protocols. Amgoud et al. (2002) also present a social semantics of communication based on argumentation. Another difference of commitment stores in comparison with interaction histories is that commitment stores have specific *commitment rules* governing the addition and removal of statements the agent is committed to. One rule may specify, for example, that if the agent retracted a previously asserted claim, it must also retract every claim based on the former via logical deduction. Another relevant concept is that of *pre-commitment* proposed by (Colombetti, 2000). A request pre-commits the utterer in the sense that the utterer will be committed in case the hearer accepts the request. Commitment stores enable us to capture such pre-commitments.

In the ABN literature, (Amgoud and Parsons, 2001) define for each agent a publicly accessible commitment store. Adding statements to the commitment store is governed by the dialogue-game rules. For example, when an agent accepts a request for action p, then p is added to its commitment store. Agents may also be allowed to retract commitments under certain conditions. In the context of purchase negotiations, (McBurney et al., 2003) dealt with the issue of retraction differently. For example, the framework involves two locutions: `agree_to_buy`(.) and `agree_to_sell`(.), for committing to certain resource exchanges. Instead of providing explicit locutions for retracting

[7]Sierra et al. (1998) use the term *negotiation thread*, while (Sadri et al., 2001b) use the term *dialogue store*.

these commitments, the authors provide additional locutions: `willing_to_buy`(.) and `willing_to_sell`(.), which are softened versions of the former locutions, however, with no commitments incurred (i.e., they are free to refuse to sell or buy something they have previously agreed upon). This way, agents may usefully provide information without necessarily committing to it or having to explicitly retract it.

Finally, Bentahar et al. (2004) have recently proposed a dynamic logic approach to capturing commitments in argumentation. Their model explicitly relates the different actions of the agents to the commitments that ensue. Thus, as a result of different actions, the commitments reach different states (e.g. active, withdrawn, satisfied) in a way similar to (Fornara and Colombetti, 2003). Hereunder we provide an example from (Bentahar et al., 2004) of an illocution creating a particular commitment:

$$M, Pa, s_i \models \textit{Accept-content}(Ag_2, SC(Id_0, Ag_1, Ag_2, \varphi)) \text{ iff}$$

$$M, Pa, s_i \models \textit{Active}(SC(Id_0, Ag_1, Ag_2, \varphi)) \wedge \textit{Create}(Ag_2, SC(Id_1, Ag_2, Ag_1, \varphi))$$

where M is the Kripke model which structures the states of the world, Pa is the infinite sequence of states of the world, s_i is the given state of the world, Id_0 and Id_1 are identifiers for the different commitments captured by the predicate SC.

The above formula indicates that the *acceptance* of the content φ of the commitment by agent Ag_2 is allowed iff: (i) the SC is active (i.e. the agent has made a commitment to φ conditional upon the reception of an accept) because we cannot act on a SC content if the SC is not active and (ii) agent Ag_2 creates a SC whose content is φ. Therefore, Ag_2 becomes committed towards the content φ (which could be the result of enacting the contents of an agreement or the truth value of a particular information).

Bentahar's model forms a good basis for building practical systems for ABN agents since it connects the illocutionary contents to actual actions and the result of actions (i.e. here φ). However, their approach is limited to only considering propositions that can be attacked or challenged (in the argumentation sense) as the object of their commitments. A more expressive notion of these propositions would be needed to allow the representation of real offers that agents pass to each other during negotiation.

2.1.3.2 Challenges

The representation and manipulation of information stores is not a trivial task, and has significant effects on both the performance and outcomes of negotiation dialogues. In particular, information store manipulation rules have a direct effect on the types of utterances agents can make given their previous utterances (i.e., the protocol), the properties of the dialogues (e.g., termination), and the final outcome (e.g., the ability to change one's mind coherently).

Some of the key questions that need to be addressed in an ABN framework are: Under what conditions should an agent be allowed to retract its commitments and how would this affect the properties of dialogues? Under what conditions should an agent be forced to retract its commitments to maintain consistency? While many of these questions are being investigated in the multi-agent dialogue literature in general (Maudet and Chaib-draa, 2003), there are issues specific to negotiation dialogues. In particular, commitments to providing, requesting, and exchanging resources may require a different treatment from commitments in other types of dialogues, such as persuasion or information seeking. This is because negotiation dialogues involve selfish agents for whom the only goal is to come to an agreement that maximises their individual utility while agents in the persuasion or information exchange dialogue are more interested in ensuring consistency in the beliefs agents share in the dialogue. Thus, agents in negotiation dialogues are more likely to make commitments *conditional* upon the acceptance of an offer while agents in persuasion or information dialogues will mostly make commitments to the truth of statements they make. How these conditional commitments can be managed has received relatively less attention in the community than the commitments agents make in persuasion or information dialogues.

2.2 Elements of ABN Agents

In the previous section, we discussed the different elements of an ABN framework that are external to the participating agents. Issues such as the protocol and languages help to define the system in which agents operate, but often these say little about how agents are specified, or how they reason about the interaction.

Before we get into a discussion of the general features of an ABN agent, we shall describe what constitutes (at an abstract level) the decision making model (see figure 1.1 in chapter 1) of a basic, non-ABN-based bargaining agent. This will allow us to clearly contrast the ABN agent from other negotiators, making our analysis more focused. Therefore, we begin by presenting a conceptual model of a simple negotiator in figure 2.1. This captures, on a very abstract level, the main components needed by an agent in order to be capable of engaging in bargaining.[8] This model is not meant to be an idealisation of all existing models in the literature, but rather a useful starting point for illustrating how ABN agents are different from other types of agent.

We refer to an agent involved in bargains which largely depend on exchanging proposals as a classical bargaining agent. This agent needs to have a *locution interpretation* component, which parses incoming messages. These locutions usually contain a proposal, or an acceptance or rejection message of a previous proposal. They might also

[8]For a more detailed discussion of the conceptual architectures for negotiating agents, refer to (Ashri et al., 2003).

contain other information about the interaction, such as the identity of the sender (especially in the case of multi-party encounters). Acceptance messages usually terminate the encounter with a deal. A proposal may be stored in a *proposal database* for future reference. Proposals (or rejections) feed into a *proposal evaluation and generation component*, which ultimately makes a decision about whether to accept, reject or generate a counterproposal, or even terminate the negotiation. This finally feeds into the *locution generation* component which sends the response to the relevant party or parties. A more

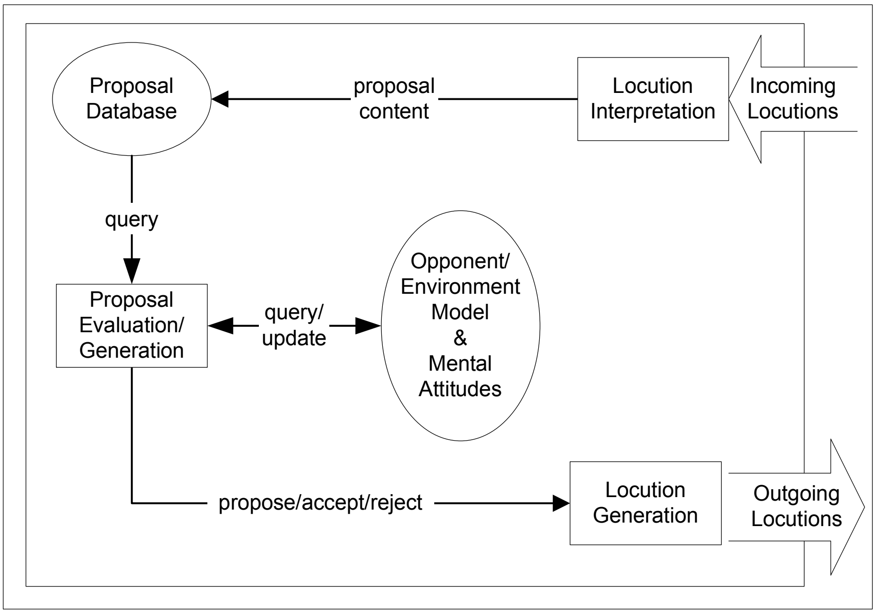


FIGURE 2.1: Conceptual Elements of a Classical Bargaining Agent

sophisticated classical bargaining agent may maintain a *knowledge base* of its mental attitudes (such as beliefs, desires, preferences, and so on (Wooldridge, 2002)), as well as models of the environment and the negotiation counterpart(s). This knowledge may be used in the evaluation and generation of proposals by judging the validity and worth of proposals made (for example, by verifying whether proposals are actually feasible and do not conflict with the current observations of the environment). Moreover, the knowledge base may be updated in the light of new information. However, the updates that can be made are somewhat limited because the only information usually available to the agent during the interaction is:

1. Proposals (or bids) from the counterpart or a competitor.

2. A message rejecting a proposal initially made by the agent.

3. Other observations from the environment (e.g., a manufacturing plant agent bidding for raw material may monitor customer demand changes and bid accordingly).

The agent may be able to infer certain things from this information. For example, by receiving a rejection the agent may infer that the counterpart does not rate certain attribute/value assignments highly. Similarly, by receiving a proposal (or by observing the proposal of another competing bidder) the agent might infer attribute values that appeal to the counterpart (or competitor), which can then guide his own bargaining or bidding strategy.[9]

In contrast with a classical negotiating agent, more sophisticated meta-level information can be explicitly exchanged between the ABN agents (see figure 2.2).[10] This, in turn, can have a direct effect on the agent's knowledge base. Therefore, in addition to evaluating and generating proposals, an agent capable of participating in argument-based negotiation must be equipped with mechanisms for *evaluating* arguments (and updating the mental state accordingly) and for *generating* and *selecting* arguments. If the

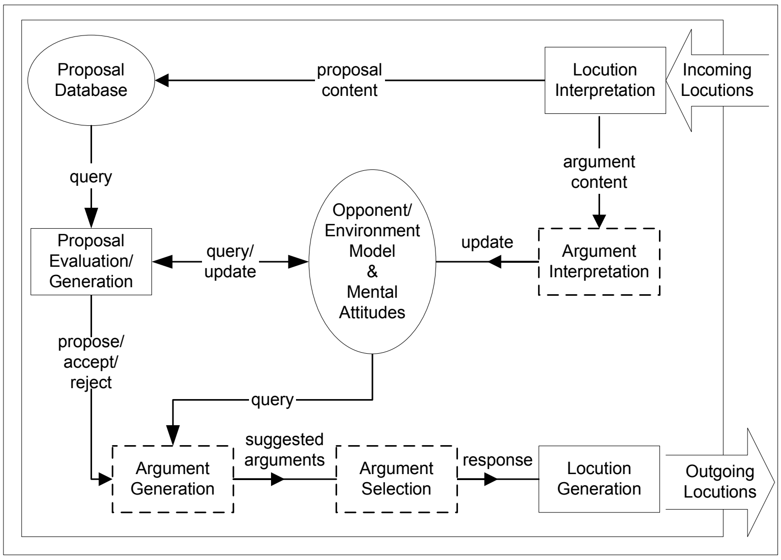


FIGURE 2.2: Conceptual Elements of an ABN Agent (the dashed lined boxes represent the additional components necessary for ABN agents).

locution contains an argument, an *argument evaluation* or *interpretation* mechanism is invoked which updates the agent's mental state accordingly. This may involve updating the agent's mental attitudes about itself and/or about the environment and its counterparts. Now, the agent can enter the *proposal evaluation* stage in the light of this new information. Note that at this stage, not only does the agent evaluate the most recent proposal, but it can also re-evaluate previous proposals made by its counterparts; these proposals are stored in the proposal database. This is important since the agent might (intentionally or otherwise) be persuaded to accept a proposal it has previously rejected.

As a result of evaluating proposals, the agent may generate a counterproposal, a rejec-

[9]Similar issues have been investigated in the study of *signalling* in game-theory (Spence, 1974).

[10]Note that the actual way in which ABN agents are designed or implemented may differ from the above. For example, the agent might perform certain operations in a different order, or might combine or further decompose certain functionalities. Therefore, our conceptual model is to be taken in the abstract sense and should not be seen as a prescriptive account of how ABN agents *must* precisely look like. Instead, it provides a useful point of departure for beginning an analysis of the generic features of these agents.

tion, or an acceptance. In addition, however, a final *argument generation* mechanism is responsible for deciding what response to actually send to the counterpart, and what (if any) arguments should accompany the response. For example, the proposal evaluation and generation component might decide that a proposal is not acceptable, and the argument generation mechanism might accompany the rejection with a critique describing the reasons behind the rejection. Such arguments might also be explicitly requested by the other party or even enforced by the protocol. Note that an agent may also choose to send a stand-alone argument (i.e., not necessarily in conjunction with a proposal, acceptance or rejection).

At times, there might be a number of potential arguments that the agent can send. For example, in order to exert pressure on a counterpart, an agent might be able to either make a threat or present a logical argument supporting some action. Deciding on which argument to actually send is the responsibility of an *argument selection* mechanism. Finally, this information is given to the *locution generation* mechanism which places this information in the proper message format and utters it.

In summary, negotiating agents must, at least, be able to:

1. interpret incoming locutions
2. evaluate incoming proposals
3. generate outgoing proposals
4. generate outgoing locutions

An ABN agent needs, *in addition*, to be able to:

1. evaluate incoming arguments and update its mental state accordingly
2. generate candidate outgoing arguments
3. select an argument from the set of available arguments

Now that we have given an overview of the features of an ABN agent, we consider each of these features in more detail. In the course of doing so, we evaluate the state of the art and outline major challenges and opportunities.

2.2.1 Argument and Proposal Evaluation

Recall that an ABN agent needs to evaluate potential agreements proposed by its counterpart(s). The agent also needs to be able to evaluate arguments intended at influencing its mental state. While proposals may be evaluated more straightforwardly through comparison with some subjective preference criteria, argument evaluation is less trivial.

Argument evaluation is a central topic in the study of argumentation, and has been studied extensively by philosophers at least from the days of Aristotle (Aristotle, 1928; Hitchcock, 2002). In Artificial Intelligence, argument evaluation and comparison has been applied, for example, in internal agent deliberation (Kakas and Moraitis, 2003), in legal argumentation (Prakken and Sartor, 2001), and in medical diagnosis (Krause et al., 1995; Fox and Parsons, 1998).

Here, however, we find it useful to distinguish between two types of considerations in argument evaluation:

1. **Objective Considerations:** An argument may be seen as a *tentative proof* for some conclusion. Hence, an agent, or a set of agents, may evaluate an argument based on some *objective convention* that defines how the quality of that proof is established. This may be done, for example, by investigating the correctness of its inference steps, or by examining the validity of its underlying assumptions. For example, (Elvang-Gøransson et al., 1993b) propose a classification of arguments into *acceptability classes* based on the strength of their construction. Arguments may also be evaluated based on their relationships with other arguments. For (Dung, 1995), for instance, an argument is said to be *acceptable* with respect to a set S of arguments if every argument attacking it is itself attacked by an argument from that set. The set S is said to be *admissible* if it is conflict free and all its arguments are acceptable with respect to S.

2. **Subjective Considerations:** Instead of applying an objective, agent-independent convention for evaluating arguments, an agent may choose to consider its own preferences and motivations in making that judgement, or those of the intended audience. In the framework presented by (Bench-Capon, 2001), for example, different participants in a persuasion dialogue have different preferences over the "values" of arguments. Argument assessment and comparison would then take place in accordance with the preferences of the dialogue participants. This means that the participants may influence the outcome of the argument evaluation process by having different *subjective* preferences.

Let us now examine the usage of the above considerations in different types of argumentation dialogues. If two agents are reasoning about what is true in the world (i.e., if they are conducting *theoretical reasoning*), then it makes sense for them to adopt an objective convention that is not influenced by their individual biases and motivations. For example, whether it is sunny outside should not be influenced by whether participants want it to be sunny, but rather only by the material evidence available.

If, on the other hand, two participants are engaged in a dialogue for deciding what course of action to take (i.e., if they are conducting *practical reasoning*), or what division of scarce resources to agree upon, or what goals to adopt, then it would make more sense for them to consider their subjective, internal motivations and perceptions, as well as the objective truth about their environment.[11] Even objective facts may be perceived differently by different participants, and such differences in perception may play a crucial role in whether or not participants are able to reach agreement. For example, a potential airline traveller may perceive a particular airline as unsafe, while the staff of the airline itself may consider it to be safe. Presumably such a difference in perceptions may be resolved with recourse to objective criteria (if any can be agreed) regarding relative crash statistics, deaths-per-mile-flown on different airlines, etc. But if, for example, potential travellers perceive a particular airline as unsafe compared to other airlines, despite objective evidence showing the airline to be safer than others, this perception may inhibit them from flying the airline anyway. The marketing team of the airline concerned, in trying to persuade potential travellers to fly with it, will have to engage in dialogue with potential customers on the basis of those customers' subjective perceptions, even though such perception may be false. For the marketers to ignore such mis-perceptions risks the dialogue terminating without the potential customers flying the airline.

In summary, agents participating in negotiation are not concerned with establishing the truth *per se*, but rather with the satisfaction of their needs. Hence, negotiation dialogues require agents to perform argument evaluation based on objective as well as subjective criteria.[12] In other words, agents need to perform *argument evaluation* as part of, or in relation to, *proposal evaluation*.

2.2.1.1 State of the Art

As we argued above, argument evaluation in negotiation must involve both objective as well as subjective considerations, and hence must involve some subjective assessment of proposals put forward by negotiation counterparts. In this subsection, we show some approaches to proposal and argument evaluation in the existing ABN literature.

One approach to proposal and argument evaluation is to assume agents are benevolent, using the following simple normative rule: *If I do not need a resource, I should give it away when asked.* This approach can be found in a number of frameworks (e.g., Parsons et al., 1998; Amgoud et al., 2000b; Sadri et al., 2001b).

Consider the following example from (Parsons et al., 1998). An agent a intending to hang a picture would produce, after executing its planning procedure, intentions to acquire a nail, a hammer and a picture. Interactions with other agents are only motivated in case

[11] Refer to (Rahwan et al., 2003d) for a related comparison between argumentation over goals and beliefs.

[12] Note that objective argument evaluation may also take into account certain "preferences", such as the trust the evaluator has in the agent proposing the argument. However, this remains aimed at establishing the truth, rather than being influenced by the agent's personal gain.

the agent is not able to fulfill its intentions on its own. Suppose the agent does not have a nail. This leads the agent to adopt a new intention (we can call that a sub-intention) to acquire a nail, which may be written $I_a(Have(a, nail))$. If a believes that another agent b has a nail, it would generate another sub-sub-intention that b gives the nail to it, written $I_a(Give(b, a, nail))$. This triggers a request to be made to agent b in the form $H_1 \vdash I_a(Give(b, a, nail)$, where the argument H_1 constitutes the sequence of deductive steps taken to reach the request.[13] In general, agent b accepts the request unless it has a conflict with it. There are two types of conflicts that would cause b to reject the request:

1. Agent b has a conflicting intention. In argumentation terms, the agent refuses the proposal if it can build an argument that *rebuts* it.

2. Agent b rejects one of the elements of the argument supporting the intention that denotes the request. In argumentation terms, the agent refuses the proposal because it can build an argument that *undercuts* it.

We shall explain the above two cases using the same picture-hanging example. An example of the first case is if agent b rejects the proposal because it also needs the nail, say to hang a mirror; i.e., it can build an argument for the intention $I_b(\neg Give(b, a, nail))$. This argument is based on (among other things) the intention $I_b(Can(b, hang(mirror)))$. An example of the second case is if, in the plan supporting the intention $I_a(Give(b, a, nail))$, agent a made the false assumption that b possesses a nail, written $B_a(Have(b, nail))$. If b actually does not have a nail, then it would adopt the intention of modifying that belief, i.e. $I_b(\neg B_a(Have(b, nail)))$. Agents continue through a process of argument exchange, which may involve recursively undercutting each other's arguments until a resolution is reached.

In order for argumentation to work, agents must be able to *compare* arguments. This is needed, for example, in order to be able to reject "weak" arguments. Parsons et al. (Parsons et al., 1998) compare arguments by classifying them into *acceptability classes* based on the strength of their construction (Elvang-Gøransson et al., 1993b). If two conflicting arguments belong to the same class, the authors assume the agent has some capability to perform comparisons based on utility analysis. However, they do not specify how this decision procedure is actually undertaken, nor do they specify the conditions it needs to satisfy.

A similar approach is taken by (Sadri et al., 2001b). This framework, however, does not involve arguing about beliefs. If an agent a receives a request for a resource, and needs that resource for achieving some goal g_a, the agent rejects the request, unless an alternative acceptable plan for achieving g_a can be produced by the counterpart, with

[13] Note that the argument (or plan) may not contain intentions only, but also belief and desire formulae about the agent and its environment. For example, in the argument H_1, agent a may state the assumption that it believes b has a nail, and so on.

a promise to provide any missing resources for that plan to a. Agents are also assumed to have some ordering over plans that allows them to choose between alternative plans.

In the frameworks of (Parsons et al., 1998) and (Sadri et al., 2001b) described above, argument and proposal evaluation take into account a very simplistic subjective rule; that is to give any resource requested that the agent does not *currently* need. While this may be useful for facilitating cooperative behaviour and making sure agents preserve their current subjective interests, it may not be suitable in open agent systems where agents may be purely self-interested and may refuse to provide any resources without something in return.

An alternative trend in proposal and argument evaluation in PN agents is to explicitly take into account the utility of the agent. The basic idea is that the agent would calculate the *expected utility* in the cases where it accepts and rejects a particular proposal. And by comparing the expected utilities in these two cases (i.e., in the resulting *states*), the agent would be able to make a decision about whether to accept or reject the proposal. In the framework of (Kraus et al., 1998), the agent makes a decision about whether to accept a request by evaluating three factors: (i) the **Collision_Flag**, which fires if the requested action conflicts with one of the agent's intentions; (ii) the **Convincing_Factor**, which is a value between 0 and 1 assigned to the argument using some ad hoc rule (e.g., an appeal to past promise is assigned value 1 if the agent believes it has actually made such promise, and assigned 0 otherwise); and (iii) the **Acceptability_Value**, which involves a numerical calculation of utility tradeoffs in the case of accepting the request versus rejecting it. However, it is not clear, from the paper, how these factors are combined to produce a final decision.

Sierra et al. (1998) introduce authority as a criteria for evaluating arguments. They present an *authority graph* imposed by a relation over agent roles. This graph can be used to specify, for each pair of agents, which agent has higher authority. The authors also propose a way of comparing authority levels of *sets of* agents. This can be used to compare arguments involving statements made by multiple agents. An argument H_1 is preferred to another H_2 if and only if the authority level of agents involved in H_1 is higher than those in H_2. As an example, the authors define a conciliatory agent, which accepts appeal-to-authority arguments regardless of the content of the justification of the appeal. This means that there would be no difference between a strong appeal and a weak (or even meaningless) one. While authority seems to be a useful factor in evaluating arguments in an organisation, it seems unreasonable to rely solely on it. There are, therefore, opportunities for combining authority with other argument evaluation techniques described earlier.

2.2.1.2 Challenges

The discussion above shows that the nature of argument evaluation depends largely on the object of negotiation and the way agents represent and update their internal mental states. For example, in the framework presented by (Parsons et al., 1998), agents are able to perform some objective argumentation over their beliefs about the availability of resources, the achievability of intentions, and so on. This allows agents to potentially modify each other's mental attitudes, which may influence their preferences. In frameworks such as those of (Ramchurn et al., 2003a) and (Kraus et al., 1998), on the other hand, evaluation is based solely on the direct comparison of expected utilities. Agents do not influence each other's beliefs, but rather exert pressure on each other by exercising their ability to influence the outcomes (for example, by making a promise or a threat). In other words, an agent would not *voluntarily modify* its position as a result of correcting its perceptions of the environment, but rather *forcedly concede* on its position as a result of pressure from its counterpart. Many opportunities exist for combining the objective (belief-based) and subjective (value-based) approaches to argument evaluation. For example, how can we combine the objective evaluation of the logical form of an argument with a subjective evaluation of its consequences based on utility, trust, authority, etc.?

Another challenge is that of providing unified argumentation frameworks that facilitate negotiation dialogues involving notions of goals, beliefs, plans, etc. (Rahwan et al., 2003d) argue that systems of argumentation designed for arguing about beliefs are not readily suitable for allowing for argumentation over goals, particularly due to the different ways conflict resolution among arguments must be dealt with. For example, there is a difference between attacking a goal by demonstrating that it is not achievable and attacking it by demonstrating that it is not useful.

Rahwan et al. demonstrate different ways in which goals may relate to their sub-goals, their super-goals and the agent's beliefs Rahwan et al. (2003c,d). This allows one to characterise different types of arguments that may be provided against a particular goal, and how they can, if successful, affect the agents' mental states. Rahwan et al. discuss other types of possible attacks, as a preliminary step to understanding the *space of possible influences* ABN agents may (or must be able to) exert in the course of dialogue.

The above approach has recently been taken up by Amgoud and Kaci (Amgoud and Kaci, 2004) where the goals are attributed a bipolar nature. This means that the agent may have some goals that it deems will bring about a positive outcome for itself while there are goals that bring a negative outcome. Those that are neither of the two types are termed goals in abeyance. The authors thus provide a reasoning mechanism that allows an agent to define acceptable arguments in terms of attack relations as in Rahwan et al.. Moreover, arguments are given a strength according to either the certainty with which the goal supported by the argument will be achieved or the weight the goal has in achieving the purpose of the agent. In this way, agents can determine which are the arguments that need to be used to evaluate or make an offer. However, it is not very clear how certainty and weight are determined, nor how agents are to share their goals without revealing their preferences in a bargaining context.

2.2.2 Argument and Proposal Generation

Another central problem in the study of argumentation is that of argument generation. This problem is concerned with generating *candidate* arguments[14] to present to a dialogue counterpart. These arguments are usually sent in order to entice the counterpart to accept some proposed agreement. Hence, in negotiation, argument and proposal generation are closely related processes.

2.2.2.1 State of the Art

In existing ABN frameworks, proposal generation is usually made as a result of some utility evaluation or planning process (Sierra et al., 1998). Sierra et al. assume agents have a means of generating proposals that increase (or maximise) their utilities. For (Kraus et al., 1998), (Parsons et al., 1998), and (Sadri et al., 2001b), an underlying planner generates a set of actions or resources needed to achieve some intention. Agents then request the actions or resources they cannot achieve or obtain on their own, from other agents. If they fail to obtain immediate acceptance, they may propose to perform an action (or set of actions) or to provide resources in return for acceptance. This may be done by just giving away what they do not need, or by measuring the utilities they lose in the exchange.

Proposals may be accompanied by arguments. In the framework of (Kraus et al., 1998), for example, agents may choose to accompany proposals with arguments generated using explicit rules. By means of an illustration, what follows is an informal description of the threat-generation rule for agent i:

> **IF**
> A request has been sent to agent j to perform action α &
> j rejected this request &
> j has goals g_1 and g_2 &
> j prefers g_2 to g_1 &
> doing α achieves $\neg g_1$ &
> doing β achieves $\neg g_2$ &
> i believes doing β is credible and appropriate
> **THEN**

[14] We leave the discussion of selecting the *best* argument to section 2.2.3 below.

> i requests α again with the following threat:
> if you don't do α, I will do β

If the rule body is satisfied, the corresponding threat will become a *candidate argument*. The agent may generate other candidate arguments, such as promises or appeals, using other rules.

As mentioned above, the frameworks of (Parsons et al., 1998), (Sadri et al., 2001b) and (Amgoud et al., 2000b) take a planning approach to proposal generation. Arguments are in fact generated in the process of proposal generation itself. In other words, an agent justifies a request by simply stating the truth about its needs, plans, underlying assumptions, and so on, which ultimately caused the need to arise. This is different from other utility-based approaches described above, where agents can, in a sense, *create* arguments, such as threats and rewards, by exploiting their abilities to influence the outcomes. Of course, there is nothing that directly prevents agents from combining the two.

As described earlier, (Rahwan et al., 2003c) provide a characterisation of the types of arguments an agent can make in relation to the goal and belief structures of its counterpart. This provides a more fine-grained portfolio of candidate arguments than those of (Parsons et al., 1998), (Sadri et al., 2001b) and (Amgoud et al., 2000b) (where only plans or promises can be put forward as arguments).

And finally, authority could also be used in argument generation. Sierra et al. (1998), for example, define a simple authoritarian agent, which always exploits its social power by threatening whenever possible. Boella et al. (2004) further this approach by defining different persuasive arguments that exploit norms of the environment. They outline three ways in which norms can be used to formulate persuasive arguments such as 'command', 'convince', and 'suggest'. For a command to be issued, an authority relationship must exist while for a 'convince', the agents must share a certain level of trust. For a suggestion to be made, the agents must only make sure that they trust the information shared rather that their opponent. This work is, however, very preliminary.

2.2.2.2 Challenges

More work needs to be done in order to provide a unified way of generating arguments by considering both objective and subjective criteria. Moreover, there is a need for a complete characterisation of the *space* of possible arguments. This is not necessarily a trivial task since in some frameworks the number of possible arguments may be infinite (say, if the framework allows for nested arguments about what may happen in the future, or nested dialogues).

More work is also needed to understand the influence of different factors, such as the bargaining protocol, authority, expected utility, honesty, etc. on argument generation. Specifically, how can authority be used in constructing an argument? Should an agent believe in an argument in order to present it? Can agents bluff? These are few of the questions that need to be answered before a complete framework for argument generation is achieved.

2.2.3 Argument Selection

Related to the problem of argument generation is that of argument selection. The question of argument selection is as follows: given a number of candidate arguments an agent may utter to its counterpart, which one is the "best" argument from the point of view of the speaker?

Note that argument selection may take place in conjunction with argument generation. An agent need not generate all possible arguments before it makes a selection of the most suitable one. Instead, the agent may only concern itself with the generation of the most suitable argument itself. In other words, the agent might prune the set of candidate arguments during the process of argument generation. Whether or not this is possible, of course, depends on the nature of the argumentation framework underlying the agent's decision making.

2.2.3.1 State of the Art

In the work of (Kraus et al., 1998), arguments are selected according to the following argument strength order, with threats being the strongest arguments:

1. Appeal to prevailing practice.
2. A counter example.
3. An appeal to past promise.
4. An appeal to self-interest.
5. A promise of future reward.
6. A threat.

The intuition is that a negotiator would progress from weak arguments up to the strongest ones. For example, there is no need to threaten the counterpart if an appeal is sufficient to persuade him/her to accept a request. The authors argue that generating appeals is less costly to the persuader than threats or rewards since the latter involve possible negative side-effects.

In other frameworks, argument generation is based on the relationships between arguments. Agents in the framework presented by (Parsons et al., 1998) provide the strongest argument possible based on the acceptability classes (e.g., a tautological argument, if possible). For (Amgoud et al., 2000b), agents compare arguments based on preferential ordering over their constituent propositions in a similar manner to that in argument evaluation (i.e., based on the argumentation system of (Dung, 1995)). In Kakas and Moraitis (2003), given a particular argumentation framework, they provide tactics to argue. These tactics tell agents how to choose responses to requests by either accepting or challenging or even refusing. Finally, for (Sadri et al., 2001b), agents may compare the costs of different alternative plans to present to the counterpart.

2.2.3.2 Challenges

The problem of argument selection can be considered the essence of *strategy* in ABN dialogues in general (provided the candidate arguments contain all possible arguments). However, there is very little existing work on strategies in multi-agent dialogues. Some work is emerging that investigates strategic move selection in persuasion dialogues (Amgoud and Maudet, 2002), as well as in inquiry and information seeking dialogues (Parsons et al., 2002, 2003). Similar work needs to be done on ABN dialogues in order to provide a sound theoretical base for potential applications. (Rahwan et al., 2003a) provide a preliminary, informal attempt at charactarising strategic factors in negotiation dialogues. In this work, strategies depend on various factors, such as the agents' goals, the interaction protocol, the agents' capabilities, the resources available to participants, and so on.

Suitable argument selection in a bargaining context must take into account information about the negotiation counterpart. In game theory, this information about the opponent is modelled by a probability distribution modelling the uncertainty of the first party regarding the counterparts' preferences which, in turn, determine its strategy. In cases where such a modelling is possible, *learning* techniques can be used to find patterns in the counterpart's behaviour and use these findings in future bargaining encounters with the same (or similar) counterpart(s). Thus, while PN as discussed in chapter 1 aims to reduce uncertainty in the actions of the agents, this learning mechanism could reduce uncertainty about the preferences of the agents (if preferences stay the same over multiple encounters). An example of the application of such learning techniques on repeated encounters include (Sandholm and Crites, 1995) who apply reinforcement learning in the context of the iterated Prisoner's Dilemma game to allow agents to better predict the patterns of behaviour of their opponents. Learning in less restricted negotiation protocols has also been investigated by (Zeng and Sycara, 1997).

In ABN, more sophisticated models of the negotiation counterparts are needed, and appropriate methods of updating these models are essential for understanding the dynamics of opponents' strategies, preferences, beliefs, etc. This is a particularly challenging task for ABN since agents may not only model the observed 'behaviour' of one another, but also the 'mental attitudes' motivating that behaviour. Another important question is whether and how such learning agents converge to better and quicker deals in multiple negotiation encounters.

2.3 Summary

Having analysed the various existing frameworks in detail in the previous sections, we now proceed to present a high-level view of what has been achieved in the field of ABN as a whole. In this way, we aim to identify those areas that we will try to tackle given our intention to design decision making models and protocols for PN agents.

To this end, in table 2.2, we compare the different existing frameworks in terms of their main characteristics.[15] Specifically, the first column describes the style of argumentation underlying the ABN framework. This covers the informal literature that motivates and provides intuitive backing of the research, as well as the formal theories underlying the specification of the framework (e.g., decision theory, argumentation theory, dialogue games, etc.). When taken together this provides an idea of the starting point of each framework. As can be seen, frameworks such as those of (Amgoud et al., 2000b) and (Sadri et al., 2001b) start from a single-agent proof procedure and try to split it into multiple disjoint agents while preserving the correctness of the proof theory.

In contrast, frameworks such as those presented by (McBurney et al., 2003) and (Rahwan et al., 2003c) start by discussing the different types of interaction patterns needed among agents, and from there attempt to create a dialogue system. In comparison to these logic or mentalistic based ABN mechanisms, relatively little work has been carried out on PN mechanisms to the exception of (Kraus et al., 1998) and (Sierra et al., 1998).

The next column describes the protocol. It is clear that some ABN frameworks have not yet addressed the protocol definition, while in others it is the mainstay of their contribution. Moreover, the frameworks can differ in the way they specify the protocols, by making them implicit or explicit, defining them as finite state machines, as dialogue games, and so on. As can be seen, only (Sierra et al., 1998) defines a protocol for PN, but their approach is only limited to defining the participation rules and does not define what commitments agents make during a dialogue. The third column describes some of the important assumptions that each framework makes. In some frameworks, for instance, the agents must be cooperative for ABN to work. Frameworks can also vary

[15] Wherever the framework in question has not addressed the particular attribute of the table (e.g. Protocol) significantly, we note this as N/A.

in their assumptions about agents' utilities and preferences. Finally, we have specified whether the framework has been implemented and, if so, what form this takes.

In table 2.3, we outline the various frameworks in terms of whether and how each framework addresses the problems of argument generation, selection, and evaluation. One important observation from this is that argument selection has had very little attention in the ABN community. This is, we believe, partly because effective strategies for deciding what arguments to utter are likely to be protocol-dependent. Consequently, there is still no formal theory of bargaining protocols covering all types of mechanisms.

As can be seen, there is a clear contrast in the way the three main mechanisms are conceived by the different frameworks. We contend that this is mainly due to the differences in the underlying style of argumentation. However, despite these differences, their contributions are broadly complementary.

In light of these comparisons, we have identified a number of problems that pervade the particular area of PN. Specifically, one of the main problems is that the semantics and constructs of arguments (i.e. what is contained in an argument) are not clearly defined. Given this, we aim to develop a precise definition of arguments (rewards in particular) in repeated encounters. Moreover, as per our stated objectives in chapter 1, we also aim to develop a complete protocol for PN, as well as the decision making model that fits competitive settings. To this end, we will extend those participation rules defined by Sierra et al. (discussed in section 2.1.2.1) and build upon the specification of commitments by Bentahar et al. to encompass PN offers and arguments (described in section 2.1.3.1). Moreover, given the lack of implemented models and empirical evaluation in this area, we propose to implement our PN mechanism and determine the negotiating agents' performance given different strategies in generating offers and arguments. In so doing, we aim to provide the first objective assessment of the role of arguments in enhancing bargaining.

Authors	Characteristic			
	Argumentation style	Protocol	Main Assumptions	Implementation
Kraus, Sycara and Evenchik	- Based on psychology of persuasion (Karlins and Abelson, 1970) - PN	- Implicit in agent specification	- Agents have utility function - Agents share same architecture	- Blocks World Scenario (implemented)
Sierra, Jennings, Noriega, and Parsons	- Same as above	- Finite State Machine - Allows passing generic meta information	- Argument generation, selection and evaluation are predefined - E-Insitution present	- N/A
Parsons, Jennings and Sierra	- Logic-based, inspired by (Elvang-Gøransson et al., 1993a)	- Finite State Machine - Allows passing generic meta information	- Agents are cooperative - Agents share same architecture	- N/A
Amgoud, Maudet and Parsons	- Classification of dialogues based on (Walton and Krabbe, 1995) - Logic-based	- Dialogue Game protocol, allows sending promises, challenges, assertions, and requests	- Agents share same architecture - Complete preferences over knowledge bases - Agents share preferences	- N/A
Sadri, Torroni and Toni	- Based on Abductive Logic Programming - View of dialogue from (Walton and Krabbe, 1995)	- Implicit in agent specification	- Agents share same architecture - Agents are cooperative	- Logic agents negotiate to exchange tools
McBurney, van Eijk, Parsons, Amgoud	- View of negotiation based on (Walton and Krabbe, 1995) - Agent model influenced by consumer modelling models in marketing(Lilien et al., 1992)	- Dialogue game protocol allows passing potential offers, preference statements	- Agents do not decide as a group, goods purchased afford negotiation - Agent has utility function and other agents might know its valuation for goods	- N/A
Rahwan, Sonenberg, and Dignum	- Approach influenced by interest-based negotiation among humans (Fisher and Ury, 1983)	- Sketched locutions	- Agents have comparison criteria for selecting goals based on their support	- N/A

TABLE 2.2: Characteristics of Extant ABN models

Authors	**Mechanism**		
	Generation	Selection	Evaluation
Kraus, Sycara, and Evenchik	- Rules determine possible arguments from available pool	- Rigid order among argument *classes* only (start with appeals, then rewards, then threats)	- First using rules, then taking the conflicting intentions into account
Sierra, Jennings, Noriega, and Parsons	- Partially specified by roles	- N/A	- Only rules based on authority are presented
Parsons, Jennings, and Sierra	- Starting from existing intentions, generate tentative plans and proofs (arguments) using simple BDI planning rules	- Possibly choose strongest argument based on acceptability classes	- Based on acceptability classes - Accepted if argument cannot be logically defeated
Amgoud, Maudet, and Parsons	- N/A	- N/A	- Based on interaction between arguments and preferences over their contents
Sadri, Torroni and Toni	- Based on rules	N/A	- Compare plans with or without exchange of resources
McBurney, van Eijk, Parsons and Amgoud	- N/A	- N/A	- Informally specified
Rahwan, Sonenberg, Dignum	- Based on a list of possible attacks on goals	N/A	- Based on the effect of updating relationships between goals, supergoals, subgoals and beliefs

TABLE 2.3: Necessary mechanisms for an ABN framework

Chapter 3

Trust in Multi-Agent Systems

In the previous chapter we surveyed the ABN literature and determined the particular requirements of our PN mechanism. In this chapter, we turn to the issue of trust in order to determine the requirements of our trust model. More specifically, we evaluate the most prominent trust models from the literature at the individual level in order to determine which components of our trust model we need to develop afresh and which parts we can exploit from existing models. This analysis is carried out with particular respect to making the model applicable to both bargaining and mechanism design. Moreover, we identify those system level mechanisms that aim to elicit trust. In so doing we also lay the foundations for creating a protocol (through mechanism design techniques) that selects the most trustworthy agents in resolving conflicts.

The rest of this chapter is structured as follows. Section 3.1 analyses trust models at the individual level while section 3.2 surveys existing system level trust models. Finally section 3.3 summarises the main findings and discusses the main requirements of our trust model and their relationship to mechanism design and bargaining.

3.1 Individual-Level Trust

Here we take the viewpoint of an agent situated in an open environment trying to choose the most reliable interaction partner from a pool of potential agents and deciding on how to interact with it. As we mentioned earlier (section 1.3), there are a number of ways the agent can go about doing this:

- it could interact with each of them and learn their behaviour over a number of encounters. Eventually, it should be able to select the most reliable or honest agents from the pool or devise an appropriate strategy to deal with the less (or more) reliable ones. In this case, the agent reasons about the outcome of the direct interactions with others.

- the agent could ask other agents about their perception of the potential partners. If sufficient information is obtained and if this information can be trusted, the agent can reliably choose its interaction partners. In this case, the agent reasons about interactions that others have had with its potential partners (indirect interactions).
- the agent could characterise the known motivations of the other agents. This involves forming coherent beliefs about different characteristics of these agents and reasoning about these beliefs in order to decide how much trust should be put in them.

Given the above, we can classify trust models at the individual level as either learning (and evolution) based, reputation based, or socio-cognitive based. While the learning and evolutionary models, which we consider in section 3.1.1, aim to endow agents with strategies that can cope with lying and non-reciprocative agents, reputation models, which we describe in section 3.1.2, enable agents to gather information in richer forms from their environment and make rational inferences from the information obtained about their counterparts. Finally, in section 3.1.3 we describe socio-cognitive models which adopt a rather higher level view of trust that takes the knowledge of motivations of other agents for granted and proposes ways to reason about these motivations.

3.1.1 Learning and Evolving Trust

In this section we consider trust as an emergent property of direct interactions between self-interested agents. Here we assume that the agents will interact many times rather than through one-shot interactions. This tallies with the concept of trust as a social phenomenon that is inherently based on multiple interactions between two parties (Molm et al., 2000; Carley, 1991; Prietula, 2000; Yamagishi et al., 1998; Dasgupta, 1998). It is further assumed that agents have an incentive to defect (Dasgupta, 1998). For example, defecting in an interaction could mean that the agent does not satisfy the terms of a contract, sells poor quality goods, delivers late, or does not pay the requested amount of money to a seller. In these examples, defection could get higher payoffs for the agent defecting (e.g. the seller gets paid more than the actual value of the goods sold) and cause some utility loss to the other party (e.g. the buyer loses utility in buying a low quality product at a high price). However, defection may reduce the possibility of future interactions since the losing agent would typically attempt to avoid risking future utility losses. In contrast, if both interaction participants cooperate, we assume that they get an overall higher payoff in the long run (Axelrod, 1984). For example, a seller delivering goods on time or selling goods of a high quality may result in future purchases from the buyer. In all these cases, we are generally assuming that the agents already know the payoffs associated with each of their actions.

In most encounters the move of an opponent is not known in advance. In such competitive interactions (discussed in section 1.1), the safest (i.e. minimising possible loss), and not necessarily the most profitable for the interacting agents, move will be chosen unless there can be some way to ascertain that the other party can be trusted[1]. Thus, if an agent believes its counterpart is reciprocative, then the former will never defect, otherwise it will, and both could end up with lower payoffs than if they trusted each other or learnt to trust each other. This belief may only be acquired if the game is repeated a number of times such that there is an opportunity for the agents to learn their opponent's strategy or adapt to each other's strategy.

To this end we will first consider models that show how trust, through reciprocation (of positive deeds), can be learnt or evolved over multiple direct interactions (section 3.1.1.1). These interaction models, however, greatly simplify the interactions to extreme notions of cooperation and defection. In reality we believe these two extremes can rather be considered the two ends of an axis measuring the success of the outcome of the interaction. In this context, cooperation could mean, for example, that a seller actually delivers some goods (rather than not delivering at all), but some slight delay in the delivery might still be considered poor cooperation (rather than complete defection). Hence the perception of an agent of another party's trustworthiness is relative to the level of satisfaction of the outcome. We therefore consider, in section 3.1.1.2, how the payoffs in the individual interactions can actually be modelled in realistic applications.

3.1.1.1 Evolving and Learning Strategies

The most common example used to illustrate the evolution of trust or cooperation over multiple interactions is Axelrod's tournaments revolving around the Prisoner's Dilemma (Axelrod, 1984). The Prisoner's Dilemma is a game involving two prisoners that have to decide whether to cooperate by not revealing their accomplice's deeds or to defect by revealing this information. The dilemma arises as a result of each other having to separately (in different rooms) decide to cooperate or not, resulting in some years of imprisonment (5 for one cooperating and 1 for the one defecting, 3 for both if they both defect and 1 for both if they both cooperate). In the face of such uncertainty the best strategy proves to be defection even though this does not lead to best outcomes (hence the dilemma). Within very controlled settings, Axelrod's tournaments have shown that the tit-for-tat strategy was the most successful (reaping higher average points over all the encounters) relative to other selfish or nicer (i.e. mostly cooperative) strategies. Tit-for-tat cooperates on the first move and imitates the opponent's move in the remaining interactions. By adopting this strategy, agents are, in fact, trusting each other but would punish untrustworthy behaviour if it ever happens (and also forgive if trustworthy behaviour is shown again). If two agents adopt tit-for-tat (or permanently cooperative

[1]The moves chosen will also be dependent on the risk attitude (risk seeking, risk neutral, or risk averse) of the agent. In this respect, we conceive of trust as a means to reduce the risk perceived by the agent (Yamagishi et al., 1998; Molm et al., 2000; Dasgupta, 1998).

strategies) it is shown that they end up with the highest payoffs compared to all other strategies. However, when faced with other selfish strategies, tit-for-tat does not get the maximum payoff, though it actually gets a higher payoff than most other strategies. This is because tit-for-tat actually loses on the first encounter.

It is therefore required that an agent adapts its strategy according to the type of environment (agents therein) it encounters in order to minimise loses and foster cooperation. By allowing agents to adapt, Wu and Sun have shown that trust can actually *emerge* between them (Wu and Sun, 2001). This means that the agents evolve a trusting relationship (i.e. a cooperative stratetgy) by evaluating the benefit of each possible strategy over multiple interactions. A multi-agent bidding context, in which a number of seller agents bid for contracts in an electronic marketplace, is chosen to exemplify the concept. It is first shown that when agents are all *nice* (always cooperating) to each other, sellers tend to learn to exploit them. To counter this, the nice agents learn to use tit-for-tat to minimise their losses. As a result, the nasty sellers (exploitative agents) then learn to be reciprocative since cooperating would bring them more benefit than defecting in the long run. Thus, trust emerges as a result of the evolution of strategies over multiple interactions. This example also shows that the evolution of strategies allows nice agents to beat nasty ones in the long run. However, while strictly applying to the bidding context, Wu and Sun's model does not take into account the fact that there might be some utility loss (in the short run) in cooperating with the other party (e.g. giving away some resources).

In this respect, while acknowledging a cost to cooperation, Sen[2] demonstrates how reciprocity can emerge when the agents learn to predict that they will receive future benefits if they cooperate (Sen, 1996). In a more recent set of experiments, Sen and Dutta give clear guidelines about evolutionary stable strategies (Sen and Dutta, 2002) (not necessarily tit-for-tat) in different types of environments (with different sorts of strategies). They show that collaborative liars (collaborating defectors) perform well whenever the number of interactions is small and the number of philantropic agents (always cooperating) is large. However, reciprocative strategies perform better in all other scenarios they tested. Besides proving that reciprocation pays, these results show that the length and number of interactions matter when it comes to evaluating another agent's trustworthiness. If the number of interactions is too low, then trust cannot be built. This is corroborated by Mui et al. and Wang et al. in their probabilistic trust model which identifies a threshold for the number of encounters needed to achieve a reliable measure of an opponent's trustworthiness based on performance appraisal (Mui et al., 2002; Wang and Vassileva, 2003).

In the case where this threshold cannot be reached, other techniques must be used to elicit trustworthiness. In this respect, Mukherjee et al. have shown how trust can be

[2]For a wider reading on the problem of learning cooperative strategies in competitive settings, see (Mukherjee et al., 2001; Biswas et al., 2000; Sen, 1996).

acquired if agents know their opponent's chosen move in advance (Mukherjee et al., 2001). They show that, in the case where the agents do not reveal or only partially reveal (only the first mover does) their actions before their opponent acts, no amount of trust is built since it is optimal for the opponent to always choose to defect. However, in the bilateral information revealing scenario (both agents reveal their actions), both agents trust each other through *mutually learning* to choose an action that results in higher outcomes than predicted for the non-learning situation. It is to be noted that their model (as well as Sen's), besides assuming a static environment, uses an arbitrarily defined function to calculate the cost of interacting and returns from future actions (the basis of which might need more investigation but has proven to be quite successful in the applications that have been simulated).

Up to this point, all the above models deal strictly with the problem of cooperation between self-interested parties. However, not all multi-agent interactions are strictly competitive. For example, agents may be self-interested, but still need to achieve a maximum payoff as a group or society since the latter determines their individual payoffs (e.g. individuals contributing an unspecified amount of money to build a road in their community such that the total amount collected decides whether the road will be built, giving utility to the individuals, otherwise the money is used for a secondary purpose). This is the problem tackled by Birk (Birk, 2000, 2001). It is thus shown that trust may not only emerge from the evolution of strategies (Birk, 2000), but can also arise strictly out of learning (Birk, 2001). The learning method Birk exposes uses a continuous case N-prisoner's dilemma as basis for simulation. This involves agents contributing to a common fund required for the society to achieve its goals, but each agent is tempted to contribute less than the equal split of the total investment required, in the hope that others will contribute more. In this context, a cooperative strategy (i.e. contributing more than the equal split) gradually predominates in an environment where bad agents (i.e. contributing less) are in the majority. This is because the low investment obtained by the society impacts negatively on the utility of each individual member as well, forcing the latter to learn to cooperate to get higher payoffs. However, as the number of cooperative agents increases, the agents learn to defect again to get better payoffs (this is similar to what Wu and Sun's model predicts). Birk's results additionally show that the society reaches an equilibrium with a high level of trust (or cooperation) among its members.

The above learning and evolutionary models of multi-agent strategic interactions assume complete information (e.g. strategies, payoff matrix) for the multi-agent learning algorithms to work. These results have typically been obtained through simulations using very strict assumptions and static settings (as opposed to learning in dynamic settings as in Banerjee and Peng (2004)) rather than real life scenarios where the main assumption of complete information about payoffs simply does not hold. Also, most of the learning models conceive the outcome of interactions as being bistable, that is, either a defection or cooperation. To be more realistic, we believe agents need to infer, from the information gathered through their direct interactions, how their opponents are performing and how their performance is affecting their goals. This leads on to devising realistic trust metrics.

3.1.1.2 Trust metrics

For an agent to computationally model its trust in its opponent, it is first required that the former can ascribe a rating to the level of performance of its opponent. The latter's performance over multiple interactions can then be assessed to check how good and consistent it is at doing what it says it will. Therefore, in addition to a performance rating, an agent also needs a means of *keeping track* of the performance of an agent (in its direct interactions with it). Marsh was the first to model trust computationally. His ideas on trust had their roots mostly in sociology and economics (Marsh, 1994). He modelled trust according to the subjective perception of one agent about another. Specifically, he views trust as a 'situational' concept whereby an agent would trust another differently in different situations (given by the risk and the importance of an issue that an agent is to be trusted about). In his model, trust is a value between -1 and 1 and for each variable the agnt is to be trusted upon, the trust value is weighted by variable's importance. This, in turn, means that the trust value is not significantly altered whenever the agent defects, and could lead the latter to exploit its opponent over more interactions (see chapter 5 for more details).

More recently, Witkowski et al. proposed a model whereby the trust in an agent is calculated based on its performance in past interactions (Witkowski et al., 2001) (the context is a trading scenario for an intelligent telecommunications network where bandwidth is traded, the quality and quantity of which is varied depending on the trust suppliers and buyers have in each other). The update to the trust value is different for the different types of agents defined in the system. Specifically, consumers update their trust value according to the difference between their bids and the received goods (bandwidth in this case). The better the quality (size) of the goods the higher the increase in trust and conversely for low quality goods. A higher trust in a seller would then result in it being chosen for future purchases (conversely for low trust). In contrast, the supplier agents update their trust in the consumers according to the extent to which the quality (size) of the goods (bandwidth) supplied has been exploited. If the quality offered was not fully used, then the trust goes down since it implies that the consumer has dishonestly asked for more than it actually needed. If the quality is fully exploited, the trust goes up. Results of the experiments show how trust (of consumers in suppliers) is effectively strongly dependent on the ability of suppliers to cope with the demand.[3]

[3]It is to be noted, however, that their model increases an agent's trust even if the performance of its opponent has not been faultless (e.g. a buyer not using the bandwidth completely but partially). This allows the opponent to exploit the agent so long as the opponent is not "completely defecting".

The model used by Witkowski et al. simplifies the calculation of trust through equations that deal with measurable quantities of bandwidth allocation and bandwidth use. Other models such as (Mui et al., 2002; Sen and Sajja, 2002; Schillo et al., 2000) consider the performance of an agent to be simply a bistable value (good or bad). While these models achieve the objectives of the agents for the specific simulation settings studied, they cannot generally be used more widely because realistic interactions in an open distributed system involve richer outcomes (e.g. quality of goods traded, efficiency of task handling, duration of task). To overcome this, we need more generic means of assessing performance over time. To this end, Sabater et al. (through the REGRET system) do not just limit the overall performance to a bistable value or to an efficiency measure (as per Witkowski et al.), but rather attribute some fuzziness to the notion of performance (Sabater and Sierra, 2002). Thus, depending on the context, the performance of an agent can be subjectively judged on a given scale where -1 represents very poor performance, 0 represents neutral, and +1 represents being very good. REGRET actually gives richer semantics to ratings (or *impressions*) by defining their particular characteristics. For example, an agent can express a satisfaction -0.5 for the delivery date of some goods and +1 for the price of the same goods. These impressions are then analysed and aggregated using fuzzy reasoning techniques to elicit a representative value for the overall impression (or trust) of one agent on another.

In contrast to Witkowski et al.'s model, REGRET's evaluation of trust is not only based on an agent's direct perception of its opponent's reliability, but it also evaluates its behaviour with other agents in the system. This is carried out because only perceiving direct interactions can pose a number of problems. For example, in an open system, it would be very difficult for an autonomous agent to select an interaction partner if the agent itself had never interacted with another party (i.e. it has no history to analyse). Moreover, the method opens itself to attack by strategic liars which, knowing how they are rated by the other side, can adapt their behaviour (e.g. clients overloading their channels) to make the other party believe it is trustworthy (i.e. fully using its bandwidth). In such cases an agent could be better off evaluating other environmental parameters (such as asking other agents about their impressions on each other) in an attempt to get a more reliable rating of its opponents. However, a number of problems arise in doing this. For example, information gathered from other agents could be wrong or incomplete. Such problems are exemplified and studied in section 3.1.2.

3.1.2 Reputation Models

Reputation can be defined as the opinion or view on someone about something (Sabater and Sierra, 2002). Here we consider that this view can be mainly derived from an

While this property of the model may not harm the system analysed by the authors, it seems to be counterintuitive to the ideal attributes of a trust model which should prevent the agent implementing it being exploited.

aggregation of opinions of members of the community about one of them[4]. In multi-agent systems, reputation can be useful when there are a large number of agents interacting (e.g. online auctions, stock-trading). Reputation should, for example, enable buyers to choose the best sellers in the system (e.g. on eBay, the buyers rate the sellers they interact with and this rating is provided to future buyers for them to choose the most reliable seller(s)). Moreover, reputation can induce sellers to behave well if they know they are going to be avoided by future buyers as a result of their reputation going down due to bad behaviour. These different aspects of reputation divide the field into the following lines of research:

- devising methods to *gather ratings* that define the trustworthiness of an agent, using relationships existing between members of the community.
- devising reliable reasoning methods to gather as much information from the *aggregation of ratings* retrieved from the community.
- devising mechanisms to *promote ratings* that *truly* describe the trustworthiness of an agent.

The last of the above items is dealt with in section 3.2.2 (since it falls within the realm of system-level trust). For now we will be concerned with the first two items because these are at the level of individual agents.

In order to organise the retrieval and aggregation of ratings from other agents, most reputation models borrow the concept of a *social network* from sociology (Burt, 1982; Buskens, 1998). Similar to human societies, this assumes that agents are related to each other whenever they have roles that interconnect them or whenever they have communication links (e.g. by observation, direct communication, or as information sources) established between one another. Through this network of social relationships, it is assumed that agents, acting as *witnesses* of interactions, can transmit information about each other (Panzarasa et al., 2001). Information takes the form of a performance rating (e.g. good or bad, seller delivers late, buyer never paid) as explained in the section 3.1.1.2. Such a rating could then be shared by the different nodes of the social network, thus giving rise to the concept of reputation.

3.1.2.1 Retrieving Ratings from the Social Network

Yu and Singh tackle the problem of retrieving ratings from a social network through the use of *referrals* (Yu and Singh, 2002a). In this context, referrals are pointers to other

[4]We here distinguish between trust and reputation in the sense that the former is derived from direct interactions while the latter is mainly acquired (by an agent about another) from the environment or other agents and ultimately leads to trust. This distinction is only made to facilitate the study of different models presented, rather than to prescribe such an approach to trust and reputation.

sources of information similar to links that a search engine would plough through to obtain a web page or url. Through referrals, an agent can provide another agent with alternative sources of information about a potential interaction partner (particularly if the former cannot handle the latter's request itself). Yu and Singh propose a method of representing a social network (based on a referral network (Singh et al., 2001)) and then provide techniques to gather information through the network (Yu and Singh, 2003). Specifically, they show how agents can explore a network by contacting their neighbours and can use referrals gathered from the latter to gradually build up a model of the social network. Furthermore, Schillo et al. enrich the representation of an existing social network by annotating nodes of the network to represent their particular characteristics (Schillo et al., 2000). Thus each node of the network holds two values: (i) the trust value which describes the degree of *honesty* of the agent represented by the node, and (ii) the degree of *altruism* (i.e. being good to others even at the expense of one's own utility). Both of these values are used to deduce the trustworthiness of witnesses queried at the time of calculating the reputation of potential interaction partners (see section 3.1.2.2). From an established social network it is then possible to derive higher level concepts. For example, Sabater and Sierra (Sabater and Sierra, 2002) and Yu and Singh (Yu and Singh, 2002a) derive the concept of a *group* or *neighbours* from the social network by identifying those nodes (agents) that are close together (linked together). Thus, having a social network represented allows an agent to select and contact those agents it needs in order to get a proper measure of the reputation of another agent. For example, Yu and Singh's model takes into account ratings from those agents that are close (by virtue of the number of links separating them with a potential interaction partner) to choose witnesses for a particular agent. Underlying this is the assumption that closer witnesses will return more reliable ratings.

It is further assumed, in all of the above models, that witnesses share ratings freely (i.e. without any profit). This is a relatively strong assumption which can be removed if proper mechanisms are implemented (as will be seen in section 3.2.2). Therefore, given that agents have represented their social network and properly extracted the ratings of their counterparts from the network, they then need to aggregate these ratings so as to form a coherent impression of their potential interaction partners.

3.1.2.2 Aggregating Ratings

Several means of aggregating ratings in online communities already exist. For example, in eBay (eBay, 2003), ratings are +1 or -1 values (in addition to textual information) that are summed up to give an overall rating. Such simplistic aggregation of ratings can be unreliable, particularly when some buyers do not return ratings (see (Kollock, 1999; Resnick and Zeckhauser, 2002) for a complete account of online reputation systems). For example, a sum of ratings is biased positively when there are less people not reporting bad ratings even though these people have had bad experiences. Having no rating is not considered as a bad rating, nor as a good rating and is simply discarded from the aggregation. Moreover, ratings are open to manipulation by sellers trying to build their reputation. While the latter problem can be dealt with by designing sophisticated reputation mechanisms (see section 3.2.2), the former problem can be solved at the level of the agent's reasoning mechanism.

To this end, Yu and Singh deal with absence of information in their reputation model (Yu and Singh, 2002b). The main contribution of their work is in aggregating information obtained from referrals while coping with the lack of information. More specifically, they use the Dempster Shafter theory of evidence to model information retrieved (Yager et al., 1994). The context is the following: an agent may receive good or bad ratings (+1 or -1) about another agent. When an agent receives no rating (good or bad), how should it classify this case? In Yu and Singh's model, a lack of belief (or disbelief) can only be considered as a state of uncertainty (where all beliefs have an equal probability of being true). Dempster's rule allows the combination of beliefs obtained from various sources (saying an agent is trustworthy, untrustworthy, or unknown to be trustworthy or not) to be combined so as to support the evidence that a particular agent is trustworthy or not. Moreover, together with a belief derived from ratings obtained, an agent may hold a belief locally about the trustworthiness of another due to its direct interaction with it. However, in such cases, the ratings obtained from witnesses are neglected. Nevertheless, their measure of reputation does not discredit nor gives credit unnecessarily to agents (as eBay does) in the absence of information. Another recent approach taken by Wang et al. (Wang and Vassileva, 2003) is to capture the performance of agents interacting in peer-to-peer systems according to a Bayesian network. In such a network it is possible to attribute different levels of reputation in each particular aspect of an agent's capabilities (e.g. an agent might be good in delivering files at a high speed while being very bad at delivering good quality files). In this paper it is shown how agents can better their performance by inferring information from all other agents' Bayesian networks. However, the percentage improvement in using such a probabilistic approach is not very high given the number of interactions it takes to build the bayesian network accurately.

As can be seen, Yu and Singh and Wang et al. do not deal with the possibility that an agent may lie about its rating of another agent. They assume all witnesses are totally trustworthy. However, an agent could obtain some benefit by lying about its rating of an opponent if it is able to discredit others such that it appears to be more reliable than them. In this respect, Schillo et al. deal with the problem of lying witnesses (Schillo et al., 2000). They first decompose the rating into social metrics of trust and altruism (see section 3.1.2.1). The latter metrics are used in a recursive aggregation over the network taking into consideration the probability that the witnesses queried may lie to (or *betray*) the querying agent. In this way, the value obtained for the trust in an agent is more reliable than fully trusting witnesses as in the case of Yu and Singh's model (which assumes cooperative settings). The probability of a witness lying to the querying agent is actually learnt over multiple interactions in Schillo et al.'s model. Similarly, Sen et al. extend this work and demonstrate how agents can cope with lying witnesses in their environment through learning rather than attributing subjective probabilities to the event of a witness lying (Sen and Sajja, 2002; Sen et al., 2000). Specifically, they develop a reputation model which makes the same simplifying assumptions as those illustrated in section 3.1.1. Their approach shows how the sharing of trust values (or reputation) can benefit reciprocative agents in the long run. In the short run though, selfish and lying agents still benefit from totally reciprocative agents. Furthermore, it is shown that, over time, colluding agents cannot exploit reciprocative agents if these learn the behaviour of the former and share their experience with others of a similar type. The reciprocative agents then become selfish towards these lying and completely selfish agents so as to minimise utility loss in interacting with them. Their model, however, fails when the number of witnesses in the environment falls below a given threshold. This is because a sufficiently high number of witnesses is needed to report ratings about most lying agents in population. If this is not the case, there is a higher probability of a reciprocative agent interacting with a lying one which has not previously been encountered by the witnesses.

While Yu and Singh's model demonstrates the power of referrals and the effectiveness of Dempster Shafter's theory of evidence in modelling reputation, Schillo et al.'s, and Sen and et al.'s models show how witness information can be reliably used to reason effectively against lying agents. These models, however, greatly simplify direct interactions and fail to frame such interactions within the social setting (i.e. relative to the type of relationships that exist between the witnesses and the potential interaction partners). To overcome this limitation, Sabater and Sierra adopt an (sociological) approach closer to real life settings (Sabater and Sierra, 2002). Thus their reputation value, which is representative of the trust to be placed in the opponent, is a weighted sum of subjective impressions derived from direct interactions (the *individual dimension* of reputation), the group impression of the opponent, the group impression on the opponent's group and the agent's impression on the opponent's group (together, all of these compose *the social dimension* of reputation). Now, the weights on each term allow the agent to variably adjust the importance given to ratings obtained in these diverse ways. Moreover, older ratings, devised as shown in section 3.1.2.1, are given less importance relative to new ones. The strong realism of REGRET also lies in its definition of an *ontological dimension* that agents can share to understand each other's ratings (e.g. a travel agent being good might imply low price for one agent, but may imply good quality seats reserved for another). However, REGRET does not handle the problem of lying (strategically) among agents. Ratings are obtained in a cooperative manner (from an altruistic group) rather than in a competitive setting (where witnesses are selfish). Moreover, the aggregation method REGRET uses can be sensitive to noise since ratings are simply summed up.

Some recent work in tackling noisy ratings has been proposed by Whitby et al. (Whitby et al., 2004). In this paper, it is shown how noise, in the form of unfair ratings, can be filtered out from the reputation system. Thus it is shown that by excluding the percentage of reports that fall out of the general characterisation of a particular agent, a more accurate measure of the agent's reputation can be obtained. Note that this system is based on Bayesian networks which captures the ratings of each agent about each other. Other approaches in a similar vein include (Josang and Ismail, 2002) and (Wang and Vassileva, 2003).

3.1.3 Socio-Cognitive Models of Trust

The approaches to modelling trust at the individual level that we have considered in the previous sections are all based on an assessment of the *outcomes of interactions*. For example, learning models consider the payoffs of each individual strategy, while reputation models assess outcomes of both direct and indirect interactions (i.e. third-party assessments). However, in assessing the trustworthiness of an opponent, it may also be important to consider the subjective perception[5] on the latter since it enables a more comprehensive analysis of the characteristics of the opponent (Dasgupta, 1998; Gambetta, 1998). For example, the tools and abilities available to that opponent could be (subjectively) assessed to check whether or not the agent can indeed use these to carry out an agreed task. Such beliefs or notions are normally stored in an agent's mental state and are essential in assessing an agent's reliability in doing what it says it will (i.e. being capable), or its willingness to do what it says it will (i.e. being honest).

In this respect, we report the line of work initiated by Castelfranchi and Falcone (Castelfranchi and Falcone, 1998, 2000b,a). In particular, they highlight the importance of a cognitive view of trust (particularly for Belief-Desire-Intention agents (Wooldridge, 2002)) in contrast to a mere quantitative view of trust (sections 3.1.1 and 3.1.2).

The context they choose is that of task delegation where an agent x wishes to delegate a task to agent y. In so doing agent x needs to evaluate the trust it can place in y by considering the different beliefs it has about the motivations of agent y. They claim the following beliefs are essential (in x's mental state) to determine the amount of trust to be put in agent y by agent x (these have been adapted and summarised):

- *competence* belief: a positive evaluation of y by x saying that y is capable of carrying out the delegated task as expected. If agent y is not capable, there is no point in trusting it to accomplish the task fully.
- *willingness*[6] belief: x believes that y has decided and intends to do what it has

[5]By subjective, we mean that these beliefs are formed according to the assessment of the environment and the opponent's characteristics which could also include an analysis of past interactions.

[6]In order to have this belief, agent x needs to model the mental attitudes of agent y.

proposed to do. If agent y is not believed to be willing to do the task, it might be lying if it says it wants to do so. This would then decrease x's trust in y.

- *persistence*[6] belief: x believes that y is stable enough about its intention to do what it has proposed to do. If y is known to be unstable, then there is added risk in interacting with y, hence a low trust would be put in y even though it might be willing to do the task at the point the task is delegated.

- *motivation* belief: x believes that y has some motives to help x, and that these motives will probably prevail over other motives negative to x in case of conflict. This highlights the possibility for y to defect as argued in section 3.1.1. The motives mentioned here are the same as the long term gains obtainable in helping x achieve its goals. If y is believed to be motivated (to be helpful or positively reciprocative as in section 3.1.1), then x will tend to trust it.

To devise the level of trust agent x can place in agent y, agent x would need to consider each of the above beliefs (and possibly others). These beliefs actually impact on trust, each in a different way, and these need to be taken into account in a comprehensive evaluation of all beliefs concerned. For example, the competence belief is a *pre-requisite* to trust another agent, while the motivation belief would vary according to the calculation of the future payoffs to the agents over multiple interactions. This kind of strategic consideration becomes even more important when such beliefs are known to all actors (i.e. the preferences of agents are public). For example, what could happen if agent y knows that x trusts it, or relies on it? The authors claim that this may increase the trustworthiness of x in y's mind, the self-confidence of y, or its willingness to serve x, which in turn change the trustworthiness of y. Agent x can then take into account the possible effects of its trust in y (even before performing the delegation) to support its decision of delegating. However, Castelfranchi and Falcone's approach is strongly motivated from humans which are not always rational beings (as opposed to what we expect agents to be).[7]

As opposed to the cognitive approach of Casltefranchi and Falcone, Brainov and Sandholm support the need to model an opponent's trust (as described above) with a rational approach (Brainov and Sandholm, 1999) (they specifically target the context of non-enforceable contracts). They do so by showing that if an agent has a precise estimation of its opponent's trust (in the former), this leads to maximum payoffs and trade between the two agents. However, if trust is not properly estimated, it leads to an inefficient allocation of resources between the agents involved (hence a loss in utility) since both under-estimate or over-estimate their offers on exchanged contracts. It is also shown that it is in the best interests of the agents, given some reasonable assumptions,

[7]Castelfranchi and Falcone do not show what agent y would gain in trusting x in the case presented here. If we consider rational agents to be utility maximising with respect to the goals set by their human designers, then agent y has no apparent reason to trust x more than it should if there is no gain in doing so, and it would be irrational to do so (from our definition of rationality).

to actually reveal their trustworthiness in their interaction partner (to efficiently allocate resources)!

While still in its infancy, the socio-cognitive approach to modelling trust takes a high level view of the subject. However, it lacks the rational grounding (as shown by Brainov and Sandholm) in rational mechanisms which learning and reputation models (and mechanisms) provide. In effect, the socio-cognitive approach could exploit the assessment performed by these models to form the core beliefs illustrated above. Thus, speaking generally, all the individual models of trust could contribute to a comprehensive evaluation of trust at the individual level. This would take into account strategies learnt over multiple interactions, the reputation of potential interaction partners, and finally the latter's believed motivations and abilities regarding the interaction. However, it can be computationally expensive for an agent to reason about all the different factors affecting its trust in its opponents. Moreover, as highlighted earlier, agents are limited in their capacity to gather information from various sources that populate their environment. Given these limitations, instead of imposing the need to devise trust at the individual level, it can be more appropriate to shift the focus to the rules of encounter so that these ensure that interaction partners are *forced* to be trustworthy. In this way, these rules of encounter can, at times, compensate for limited applicability of individual-level trust models (conversely, whenever the rules of encounter cannot guarantee interacting agents will be trustworthy, we might need to resort to individual-level trust models to do so).

3.2 System-Level Trust

As we mentioned earlier in section 1.2 in chapter 1, system designers usually engineer negotiation mechanisms with the intended properties of individual rationality, efficiency, and incentive compatibility. The last of these properties is the most important one with regards to trust as it implies that the system can incentivise honest behaviour from the agents. Apart from negotiation mechanisms, the system may also impose certain requirements on the behaviour of the agents or gather information about these in order to determine their level of reliability, hence their trustworthiness. Generally speaking, such requirements impose some rigidity on the system. However, these rules imposed by the system enable an agent to trust other agents by virtue of these different constraints. These constraints can be applied in a number of ways. Firstly, it is sometimes possible to engineer the negotiation protocol (as in mechanism design) such that the participating agents find no gain in utility by lying or colluding (or find a better gain in being honest). Secondly, an agent's reputation as being a liar (or truthful) can be spread by the system. Thus, knowing that their future interactions will be compromised if they are reputed to be liars (i.e. the shadow of the future in Axelrod's terms (Axelrod, 1984)), agents can be forced to act well (up to the point they leave a system). Thirdly, agents can be screened upon entering the system by providing proof of their reliability or honesty through the references of a trusted third party.

Against this background, we subdivide system-level trust[8] in terms of (i) devising incentive compatible protocols, (ii) developing incentive compatible reputation mechanisms (incentivise truthful revelation of reputation) that foster the selection of most reliable agents, and (iii) developing security mechanisms that ensure new entrants can be trusted (both honest and reliable). This is the structure that we adopt in the following subsections.

3.2.1 Truth Eliciting Interaction Protocols

In order to ensure truth-telling on the part of agents involved in an interaction, a number of protocols and mechanisms have been devised in recent years (see (Sandholm, 1999) for an overview). These protocols aim to prevent agents from *lying* or *speculating* while interacting (e.g lying about the quality of goods sold or proposing a higher price than one's true valuation for goods to be bought). They do so by imposing rules dictating the individual steps in the interaction and the information revealed by the agents during the interaction. Thus, by adhering to such protocols it is expected that agents should find no better option than telling the truth. Given the aim of this thesis, we do not wish to delve into a detailed explanation of all available protocols (i.e. the Vickrey-Clarkes-Groves or VCG class of mechanisms) that enforce truth telling and enforce them to a certain degree (see (Mas-Colell et al., 1995; Dash et al., 2003) for such a wider analysis). Rather we will focus on one such protocol (namely *auctions*, since these are the most widely used mechanism in multi-agent system applications).

There are four main types of single-sided auctions, namely the English, Dutch, First-price-sealed-bid, and Vickrey. In the English auction, each bidder is free to raise his bid until no bidder is willing to raise any further, thus ending the auction. The Dutch auction instead starts with a very high ask price and reduces it in steps until one of the bidders bids for the item and wins the auction. The first price sealed bid involves agents submitting their bids without knowing others' bids. The highest bidder wins the auction. In the Vickrey auction, the bids are sealed but the winner pays the price of the second highest bid.

In this context, the Dutch and English auctions enforce truth-telling on the part of the auctioneer (e.g. the winner and the winning price cannot be faked) since bids are made publicly (as opposed to Vickrey and First-price-sealed-bid auctions where the bids are

[8]In what follows, we distinguish system-level trust borne out of strategic considerations in building mechanisms (without necessary contractual commitments) from the control-trust mentioned in (Tan and Thoen, 2000). The latter is more concerned with the level control exercised by transaction procedures without any consideration for the particular strategic behaviour of agents in the system. We believe this is an important distinction since system-level trust is not only concerned with agents performing correctly, as in the case of control-trust, but also with incentivising them to provide information truthfully to the system and other agents.

hidden). However, the Dutch, English, and First-price-sealed-bid auctions do not ensure that the bidders reveal their *true valuation* of the goods at stake. This is because the dominant strategy in these auctions is to reveal either a lower valuation (in the case of Dutch and First-price-sealed-bid) or to bid only a smaller amount more than the current highest bid up to one's true valuation (in the case of the English auction). In contrast, the Vickrey auction does enforce *truth-telling by bidders* and is a common example of the class of VCG mechanisms. Here, a bidder's dominant strategy is to bid its true valuation since doing otherwise, given uncertainty about other bids and the final price to be paid, would result in some loss in utility. Bidding higher than its true valuation could end up with the agent paying more than its valuation and bidding lower than its true valuation could make it lose the auction altogether.

As pointed out above, one of the main weakness of the Vickrey mechanism is that it does not ensure truth-telling on the part of the auctioneer. The latter could still lie about the winning bid since bids are private and known only to the auctioneer (and obviously to each of the bidders in private, unless there is some amount of collusion). The auctioneer could thus ask for a higher price than the second highest bid (just below the highest bid) to the highest bidder. In so doing, the auctioneer reaps a higher benefit than it should without the bidders knowing. In this respect, Hsu and Soo have implemented a secure (i.e. ensuring the privacy of bids and the allocation of the goods to the true winner) multi-agent Vickrey auction scheme (Hsu and Soo, 2002). The scheme differs from the original Vickrey auction in that it involves an additional step of choosing the auctioneer from among the bidders (advertised on a blackboard). The bidders submit their encrypted bids to a blackboard. The auctioneer is selected at random from the bidders and it is given a key to access all sealed bids. Using this key, it can only compare the bids' values. Thus, the auctioneer can only determine the order of bids and allocate the second highest bid to the winner. This scheme also allows the auctioneer (also a bidder), the winner, and the second highest bidder to verify the result by using their keys to check the bids shown on the blackboard.

However, the Vickrey auction, and the other main ones stated above, are not collusion proof. This means that agents can collaborate to cheat the mechanism by sharing information about their bids. Collusion would first necessitate that the agents know each other before they place their bids and therefore arrange to place bids that do not reveal their true preferences (e.g. agents withholding their bids in a Dutch auction until the ask price has gone very low, or some bidders colluding with the auctioneer to artificially raise the ask price in an English auction to force others to pay a very high price, or bidders colluding to beat competitors in a Vickrey auction). To prevent the latter from happening, Brandt extends the work of Hsu and Soo by devising a collusion proof auction mechanism that ensures the privacy and correctness of any (M+1)st-price auction (Brandt, 2001, 2002) (i.e. an auction where the highest M bidders win and pay a uniform price determined by the (M+1)st price). In this type of auction, bids are *sealed* and the highest bid wins the auction but pays a *price determined by the auctioneer* (e.g. in the Vickrey auction the second highest price is paid). Only the auctioneer and the bidder know the highest bid. To allow bidders to verify whether the winning bid is actually the highest (hence checking the honesty of the winner and auctioneer) the protocol devised by Brandt distributes the calculation of the selling price between the individual buyers using some cryptographic techniques. However, the only other agent, apart from the seller, able to calculate the exact *value of the selling price* is the winner of the auction. The protocol also ensures that bids are binding. These conditions, combined with the fact that the protocol can be publicly verified, allow the identification of malicious bidders which would have tampered with the bids and prevent collusion from affecting a single bidder. While being very powerful, the protocol is computationally expensive for a large number of agents but works well for small numbers.

As can be seen above, most auctions are not robust to lying and collusion unless some security mechanism is added into them (i.e. using cryptographic techniques). The protocols mentioned above, besides constraining interactions, neglect the fact that the agents in an open distributed system might want to interact more than once. As was shown in section 3.1.1, reciprocative or trustworthy behaviour can be elicited if agents can be punished in future interactions or strictly prevented from engaging in future interactions if they do not interact honestly. For example, if a winning bidder in an auction has been found to have lied about its preferences, it could be prevented from accessing future runs of the auction (Brandt, 2002). If an agent knows it will lose utility in the future due to bad behaviour in the present, it will find no better option but to act in a trustworthy way. In this respect, earlier in the paper (see section 3.1.1) we have shown how agents could learn to actually adapt their strategy (reciprocative or not) in order to maximise their long term payoffs against different strategies over multiple runs of an auction.

However, as pointed out in section 1.3, open multi-agent systems allow agents to interact with any other agent in the environment. This could allow malicious agents to move from group to group whenever they are detected by a given group of agents and therefore exploit trustworthy agents as they move around. Also, note that the Vickrey auction (and the whole class of VCG mechanisms) does not aim to select or determine the most reliable agents that should be involved in a given interaction. Rather, agents are assumed to be completely reliable and this could lead unreliable agents, though honest, to be selected. In order to prevent this from happening, agents can be made to share their ratings of their opponent with other agents in the environment once they have interacted with them. Techniques to allow agents to gather ratings and aggregate those in a sensible way were presented in section 3.1.2. However, it was shown that these techniques do not consider the fact that we expect agents to share (true) ratings only if it brings them some utility. In open multi-agent systems, this can be achieved through reputation mechanisms which we discuss in the next section.

3.2.2 Reputation Mechanisms

As was seen in section 3.1.2, the reputation models described do not take into account the fact that the agents are selfish and therefore will not share information unless some benefit can be derived from doing so. Furthermore, these reputation models (e.g. REGRET or Yu and Singh's model) do not motivate the use of reputation by some agents to elicit good behaviour from other agents. These models aim to endow agents with a better perception of their opponent and do not consider the effect of doing so on an opponent when the latter is aware of it! Given these shortcomings of reputation models, reputation mechanisms consider the problem of inducing trustworthy behaviour and modelling the reputation of agents *at the system level*. Reputation mechanisms can operate through centralised or distributed entities that store ratings provided by agents about their interaction partners and then publicise these ratings, such that all agents in the environment have access to them. In this case, it is the system that manages the aggregation and retrieval of ratings as opposed to reputation models which leave the task to the agents themselves. In so doing, reputation mechanisms can be used to deter lying and bad behaviour on the part of the agents. Moreover, reputation mechanisms aim to *induce truthful ratings* from witnesses and actually make it *rational* for agents to give ratings about each other to the system (i.e. individually rationality).

More specifically, Zacharia and Maes have outlined the desiderata for reputation mechanisms particularly with regards to how ratings are aggregated and how these impact on the behaviour of the actors in the system (Zacharia and Maes, 2000). They do not propose such requirements for agent-based reputation systems per se, but as we move into agent-mediated electronic commerce (He et al., 2003), it is obvious that such mechanisms will guide agent-based reputation systems. These desiderata are listed below:

1. *it should be costly to change identities in the community.* This should prevent agents from entering the system, behaving badly, and coming out of the system without any loss of utility or future punishment bearing upon them.

2. *new entrants should not be penalised by initially having low reputation values attributed to them.* If new entrants have low reputation they are less favoured though they might be totally trustworthy. This actually makes the system less appealing to agents (with bad reputation) intending to (re-)enter the system.

3. *agents with low ratings should be allowed to build up reputation similar to a new entrant.* This allows an agent to correct its behaviour if it has been shown to be badly behaving in the past.

4. *the overhead of performing fake transactions should be high.* This prevents agents from building their own reputation.

5. *agents having a high reputation should have higher baring than others on reputation values they attribute to an agent.* This presupposes that agents with high reputation will give truthful ratings to others. However, this can be contentious if reputation determines the level of profit the agent acquires since it could lead to the creation of monopolies or cartels in the market.

6. *agents should be able to provide personalised evaluations.* This involves giving more than just a simple rating of +1 to -1 to allow a better evaluation of the reputation of another agent. For example, the REGRET system implements richer ratings that can be shared using the ontological dimension (see section 3.1.2.2).

7. *agents should keep a memory of reputation values and give more importance to the latest ones obtained.* This is needed to keep the reputation measure as up to date as possible and helps prevent an agent from building up positive reputation by interacting well and then start defecting (the last defection having a greater effect than its past good behaviour).

With respect to the above requirements, Zacharia and Maes present two reputation systems (targetted at chatrooms, auctions, and newsletters): SPORAS and HISTOS. While these are not strictly multi-agent systems, they present techniques to aggregate ratings intelligently and reflect the real performance of human users in an online community. In both cases, the aggregation method allows newer ratings to count more than older ones. SPORAS, however, gives new entrants low initial reputation values and therefore reduces their chance of being selected as possible interaction partners. This is a trade-off afforded to prevent identity switching. This is because an agent having low reputation would not be any better off by re-entering the system with a new identity. HISTOS is an enhancement to SPORAS which takes into account the group dynamics as in REGRET. In particular, HISTOS looks at the links between users to deduce personalised reputation values (i.e. taking into account the social network). This enables an agent to assemble ratings from those it trusts already rather than those it does not know. Moreover, both HISTOS and SPORAS have been shown to be robust to collusion. This is because those agents that are badly rated themselves have a diminished effect on the reputation of others and those they might want to protect. However, as the authors point out themselves, the major drawback is that users are reluctant to give bad ratings to their trading partners. This is because there is no incentive to give ratings in the first place (i.e. it is not incentive compatible).

In an attempt to make the report of agents' reputation truthful, they propose the CONFESS reputation mechanism (Jurca and Faltings, 2004). This actually builds up on their earlier work in (Jurca and Faltings, 2003b,a). In CONFESS, buyers and sellers pay a certain fee (the seller pays a listing fee while the buyer pays a participation tax) to engage in a transaction. Agents are incentivised, using these fees, to reveal the true reputation of the seller. In particular, it is shown that a buyer will find no better option than to reveal the true reputation of the seller and that the seller can lie only a limited number of times. Here, it is the use of these fees after the transactions happens that allows the system to enforce truthful revelation of the sellers' trustworthiness. However, their approach has some fundamental problems. Indeed, the way the payoff to agents is calculated disregards the fact that sellers can exploit buyers simply by keeping the goods *and* the payments for the goods while still reporting truthfully. This follows from a wrong modelling of the game tree in that type of interaction.

A better attempt at modelling a reputation mechanism was proposed by Dellaroccas (Dellarocas, 2002). He introduced 'Goodwill Hunting' (GWH) as a more realistic feedback mechanism, for a trading environment. This system:

- induces sellers of variable quality goods to truthfully reveal the quality of their goods.
- provides incentives to buyers to truthfully reveal their feedback.

The GWH algorithm uses the threat of biased future reporting of quality (of goods to be sold) in order to induce sellers to truthfully declare the individual qualities of their items. Specifically, the mechanism keeps track of the seller's 'goodwill'. This value represents the seller's honesty (about revealing its reliability). It is adjusted by the quality reported by buyers. Good reports bias goodwill positively and bad reports bias it negatively. To induce sellers to reveal the true quality of their goods, the goodwill factor is used to adjust the quality they wish to broadcast for the goods they wish to sell. Thus if the seller has low goodwill, the quality of the goods it tries to publicise will be actually shown to have a lower quality by the system.

To induce buyers to report their ratings of sellers, they are given rebates on future transactions in the system. It is then shown that, if buyers report untruthfully, they can drive out sellers of good quality goods, and therefore lose the opportunity of buying high quality goods. However, the mechanism makes several somewhat unrealistic assumptions about online markets. For example, it assumes that sellers are monopolists; that is, they are the only ones to sell a particular product (of varying quality). Also it assumes that buyers will interact with sellers only once. These are needed to simplify the analysis of the model. As the author points out, among other enhancements, it is still to be shown how the mechanism fares against strategic reporting from buyers whereby they force a seller to reduce the price of its goods by giving it bad ratings, hence damaging its reputation.

In a similar vein as GWH (i.e. using mechanism design techniques), Porter et al. (2002) proposed a mechanism that incentivises agents to reveal their *own* reliability as opposed to how they believe others to be reliable. As opposed to GWH, their fault-tolerant mechanism aims to result in efficient outcomes in a one-shot interaction (in the mechanism design sense). However, their approach is limited (as we show in chapter 6) since it does not consider the impressions other agents might have on those agents which truthfully reveal their reliability. Thus, an agent may be biased on its impression about its own reliability (e.g. a seller wrongly its goods to be the best or a mechanic believing its services to be better than what its clients deem it to be). This may, in turn, lead the mechanism to choose the unreliable agents (according to the unbiased trust values).

The reputation mechanisms detailed above and the interaction mechanisms discussed in section 3.2.1 try to enforce trustworthy behaviour by minimising the opportunity for agents to defect to gain higher payoffs (see our definition of trust in section 1.3). As has been shown, more of these mechanisms still need to be developed. In the case where interaction protocols and reputation mechanisms cannot guarantee trustworthy behaviour, there still exists a need to give agents in an open system the possibility of proving their trustworthiness and should enable other agents to recognise them as reliable interaction partners. One way this could proceed is by providing references from highly recognised sources. This is similar to the case of a job seeker providing its credentials to its potential new employer. Huynh et al. (2004) recently devised a model that aims to use references in this way. However, this approach is limited in that it does not consider sources that can be trusted by all agents. Rather it uses references from any agent in the environment. This procedure then leads to very high uncertainty in the reliability of the rating itself. Note that the process of gathering credentials is not the same as reputation building and acquisition which pertains to the recognition of an entire community. Rather, credential assessment falls mostly within the realm of network security which we discuss next.

3.2.3 Security Mechanisms

In the domain of network security[9] , trust is used to describe the fact that a user can prove who it says it is (Mass and Shehory, 2001). This normally entails that it can be authenticated by trusted third parties (i.e. those that can be relied upon to be trustworthy and as such are *authorities* in the system (Grandison and Sloman, 2000)). At a first glance, this does not completely fit with our initial definition of trust (see section 1.3), but it is certainly a basic requirement for the trust models and mechanisms described earlier to work (see sections 3.1.1, 3.1.2, 3.1.3, 3.2.1, 3.2.2). This is because these models are based on the fact that agents can be *recognised by their identity* and would therefore require authentication protocols to be implemented.

To this end, Poslad et al. have recently proposed a number of security requirements that they claim are essential for agents to trust each other and each other's messages

[9]We do not wish to give a complete account of network security mechanisms since this is beyond the scope of this thesis. Rather, we will focus on the main concepts and models that strictly pertain to multi-agent systems. For a wider reading on network security for open distributed systems see (Grandison and Sloman, 2000).

transmitted across the network linking them (Poslad et al., 2003) (i.e. to ensure messages are not tampered with by malicious agents):

- *identity*: the ability to determine the identity of an entity. This may include the ability to determine the identity of the owner of an agent.
- *access permissions*: the ability to determine what access rights must be given to an agent in the system, based on the identity of the agent.
- *content integrity*: the ability to determine whether a piece of software, a message, or other data has been modified since it has been dispatched by its originating source.
- *content privacy*: the ability to ensure that only the designated identities can examine a message or other data. To the others, the information is obscured.

The authors specify these requirements for the FIPA (Foundation for Intelligent Physical Agents) abstract architecture (FIPA, 2002). These basic requirements can be implemented by a public key encryption and certificate infrastructure (Grandison and Sloman, 2000). A digital certificate is issued by a certification authority, or CA, and verifies that a public key is owned by a particular entity. The public key in a certificate is also used to encrypt and sign a message in a way that only its owner can examine the content and be assured about its integrity. The two most popular public key models are PGP (Pretty Good Privacy) and the X.509 trust model (Adams and Farrel, 1999). The former supports a *web of trust* in that there is no centralised or hierarchical relationship between CAs, while the latter is a strictly hierarchical trust model for authentication (Grandison and Sloman, 2000). However, these authenticating measures do not suffice for open multi-agent systems to ensure that agents act and interact honestly and reliably towards each other. They only represent a barrier against agents that are not allowed in the system or only permit their identification in the system. In order to enforce good behaviour *in* the system, it is instead possible that certificates are issued to agents if these meet specific standards that make them trustworthy.

In order to achieve this, trusted third parties are needed to issue certificates to agents that satisfy the standards of trustworthiness (i.e. being reciprocative, reliable, honest). For example, agents would need to satisfy certain quality standards (e.g. products stamped with the Kitemark or the 'CE' marking are assured to conform to the British standards and the European community standards respectively) and terms and conditions for the products they sell (e.g. sellers have to abide by a 14-day full-refund return policy in the UK for any goods they sell). It is only upon compliance with these quality standards that the agent would be able to sell its products. To this end, Herzberg et al. present a policy-based and certificate-based mechanism which can assign roles to new entrants (Herzberg et al., 2000). A certificate in this work is signed by some issuer and contains some claims about a subject. There is no restriction on what claims can be. For example, there may be claims about organization memberships (company employee, etc.), capabilities of the subject, or even the trustworthiness (or reliability) of the subject in the view of the issuer.

The mechanism in (Herzberg et al., 2000) also enables a party to define policies for mapping new entrants to predefined business roles. Thus an agent can ensure that a new entrant will act according to the settings defined by its role or access rights. The role assigned to an agent carries with it a number of duties and policies it needs to abide by. If the agent undertakes the role, it is forced to abide by the given rules of good behaviour. The process of role assignment and access provision is performed in a fully distributed manner, where any party or agent may be a certificate issuer. Moreover, it is not required that certificate issuers be known in advance. Instead, it is sufficient that, when requested, an agent that issues certificates provides sufficient certificates from other issuers to be considered a trusted authority according to the policy of the requesting party. This allows distributed trust build-up among parties in an open environment (Mass and Shehory, 2001).

Mass and Shehory extend the work in (Herzberg et al., 2000) to open multi-agent systems (Mass and Shehory, 2001). Specifically, they take into account the fact that agents with reasoning or planning components can adapt their strategies rather than sticking to one strategy while maintaining their role (as discussed in section 3.1.1). This means that an agent's role does not fully constrain its actions so as to prevent it from reasoning strategically about its interactions with other agents. An agent could thus learn how to adapt its strategy according to the role it has. For example, an agent bearing the role of accountant in a system could report fictitious profits, thus benefiting its company's share price, while still satisfying its role. To prevent such strategic defection or wrong doing, the agent assigning the role to the new entrant is allowed to adjust its priorities or policy based on *results from interactions with others dynamically*. This presents a more realistic view of using trust (both at the individual and system level) to decide *how* to constrain the actions (or stragegies) of an interaction partner. Recent works on trust dynamics and formal modelling of trust relationships could also help in this context by ensuring that certain rules of trust are respected by agents interacting in the system (Liau, 2003; Marx and Treur, 2001; Gans et al., 2003).

3.3 Summary

In this chapter we have systematically analysed the issue of trust in open multi-agent systems in order to define the basic requirements of the trust model we intend to develop. In particular, we have related the different means of devising trust both at the individual level and at the system level. Given this analysis, we can now define the more particular requirements for our trust model:

- Our model needs to be able to learn the reliability and honesty of an agent over repeated encounters (as per the discussion in section 3.1.1). As we have seen in the latter section, most models use a probability based mechanism. In our model we intend to use a similar probabilistic approach but, in the case where a dynamic behaviour is perceived, we will use a window of past interactions in order to adapt the trust measure over time to the most recent behaviour of an opponent (in a similar way to (Sabater and Sierra, 2002)). Moreover, our model needs to be able to define non-bistable trust values in order to cater for agents that have a given degree of reliability. This trust value can then be used in a bargaining encounter, amongst other things, to restrict or enlarge the domain of values of issues that are negotiated (i.e. adjust the stance of the agent) whenever an opponent is not deemed completely reliable or honest. In addition, such a continuous measure of trust can provide a ranking of those agents deemed most reliable or honest. Some mechanism can then make a selection of the most trusted agents when it comes to determine the outcome of the negotiation.

- Attributes of the interaction context, such as the institution within which it takse place or the norms that agents have are not usually incorporated either at the individual level or the system level in existing models of trust. Nevertheless, we believe that modelling the context is important since it determines, to some extent, whether agents can be trusted (e.g. if the institution guarantees good behaviour or if the norms of the agent foster cooperation). Given this, in our trust model we aim to model these attributes and use them in determining the trustworthiness of an agent.

- Many trust models at the individual level already cater for the aggregation and dissemination of ratings from other agents in a society and we will assume such techniques can be used to generate reputation measures (see discussion in section 3.1.2). We will therefore focus on defining a component in our model that facilitates the combination of reputation measures from other agents with trust measures an agent has privately calculated.

- As we saw in section 3.1.2, individual level trust models, and in particular reputation models, do not enforce truth-telling on the part of other agents in the society. Thus when it comes to bargaining, it can only be assumed that the reputation measures obtained from other agents are reported truthfully. However, in many cases we believe this is unrealistic and so we endeavour to build our trust model such that it can be coupled to an interaction mechanism (such as the VCG class of mechanisms) that enforces truth-telling at the system level (see section 3.2.1). As discussed in section 3.2.2, existing reputation mechanisms that aim to do so are very limited and sensitive to biased reports. In combining measures from our trust model with such an incentive compatible mechanism, we intend to build the first efficient and individually rational reputation mechanism that is also robust to biased reports from some agents.

Against the above requirements, in chapter 5 we develop the CREDIT trust model and show how it can be used to influence an agent's negotiation stance in bargaining encounters to reduce the uncertainty. Moreover, in chapter 6, we propose a Trust-Based Mechanism (TBM) where we show how CREDIT can be coupled to the interaction mechanism in order to generate an efficient outcome with such properties as individual rationality and incentive compatibility.

Chapter 4

Formal Definitions

Having identified the main requirements of our persuasive negotiation (PN) mechanism and trust model in chapter 2 and chapter 3, we now focus on the basic formal definitions that we will use in our models. In this chapter we only provide those definitions that are common to all the models we develop in the rest of this thesis. In particular we define the contracts (or offers) that agents may devise during (or reach at the end of) a negotiation encounter and the utility function that is used to evaluate these contracts. We particularly structure the utility functions of any pair of negotiating agents such that these agents have payoffs as defined in the Prisoner's Dilemma (PD) (described in section 3.1.1).[1] We choose this particular game since it provides clear incentives to agents to defect (i.e. be unreliable or dishonest) to obtain higher payoffs (see section 3.1.1) while also providing incentives to them to cooperate in the long run (i.e. since both defecting causes both agents to obtain low utilities and both cooperating gives both the highest utilities in the long run). This aspect of the game is important for our trust model, since it is then possible to show that our model fosters cooperation from both interacting agents (as trust dictates how the interaction unfolds) and hence results in higher utilities as the trustworthiness of the agents is learnt over repeated encounters. The PD is also important in defining strategies for PN since cooperation and defection can be clearly ascribed to different types of arguments agents might use in bargaining (e.g. a reward might be a proposition to cooperate and allow the opponent to defect in the next game and vice versa for a reward that is asked). Thus, the semantics of arguments are clearly defined in terms of the action sets of the agents (since cooperation and defection are different actions agents might perform). Moreover, we believe such characterisation of the utility functions closely pictures realistic interactions where agents are normally involved in non-zero sum interactions (i.e. the agents do not necessarily gain utility at the expense of their opponent). The rest of this chapter is structured as follows. Section 4.1 provides the basic definitions about the agents and contracts while

[1] In chapters 6 and 8 we specialise the definition of the utility function according to requirements of the application.

section 4.2 provides the characterisation of utility functions which allows us to define an extended version of the PD (known as the multi-move prisoners' dilemma (MMPD)) and discusses its use in our PN model and trust model. Finally, section 4.3 summarises the main concepts defined in this chapter.

4.1 Basic Notions

Let Ag be the society of agents noted as $\alpha, \beta, ... \in Ag$. A particular group of agents is noted as $G \subseteq Ag$ and each agent can only belong to one group.[2] We conceive that agents within each group have a set of similar norms which define part of the context of interaction (e.g. all retailers in the UK agree to a 14-day return policy on all items they sell or all retailers in Spain close on Sunday). The attributes of the context are particularly useful in developing our trust model in chapter 5 as per the requirements mentioned in section 3.3. $\mathcal{T}$ denotes a totally ordered set of time points (sufficiently large to account for all agent interactions) noted as $t_0, t_1, \ldots$, such that $t_i > t_j$ if and only if $i > j$. In the following subsections, we define the main components of the negotiation object (see section 1.2) that agents use to define their offers particularly in bargaining encounters. Then we define the basic utility function used to evaluate these offers. Given these, we then structure the relationship between the negotiating agents' utility functions such that they play an extended version of the PD (which we define as the MMPD) as per the requirements of our trust model and PN model .

4.1.1 Contracts

After negotiation, agents usually come to an agreement that is normally termed a contract (offers made while bargaining also represent potential contracts and have the same structure). In this thesis, contracts are agreements about (commitments to) issues and the values these issues should have (as per section 1.2). Let $X = \{x_1, x_2, \ldots x_n\}$ be the set of potential issues to include in a contract, and the domain of values taken by an issue x be noted as D_x (for simplicity we assume that all D_x are an interval of real numbers $\mathbb{R}$). We will note that issue x takes the value $v \in D_x$ as $x = v$. Thus, a particular contract, O, is an arbitrary set of issue-value assignments noted as $O = \{x_1 = v_1, x_2 = v_2, ..., x_n = v_n\}$ where $x_i \in X$, $v_i \in D_{x_i}$, and $O \in \mathcal{O}$ which denotes the set of potential contracts. We denote by $\mathcal{O}$ the set of potential contracts. We will also note the set of issues involved in a contract O as $X(O) \subseteq X$. Given an agreed contract, two or more agents all have a (disjoint) subset of the contract to enact. For example, a seller has to deliver the goods at a given time while the buyer has to pay for the goods. Each subset of the contract allocated to an agent is superscripted by

[2] If $\mathcal{G}$ denotes a partition $\{G_1, G_2, \ldots, G_l\}$ of the society of agents into non-empty groups, then for all $G_i, G_j \in \mathcal{G}, G_i \cap G_j = \emptyset$, $\bigcup_i G_i = Ag$.

the respective agent identifier such that, for example, in a contract O between α and β, $O^\alpha \cup O^\beta = O$.

4.1.2 Utility Functions

We capture the preferences of an agent through its utility function (von Neuman and Morgenstern, 1944). This function outputs a measure of how much an agent prefers a particular outcome. Thus, an agent, α, has a utility function for contracts, noted as $U^\alpha : \mathcal{O} \to [0,1]$, and for each issue $x \in X(O)$ in a contract noted as $U^\alpha_x : D_x \to [0,1]$. We will generally assume that the utility functions are *linear* so as to simplify the analysis of the properties of the models we study and facilitate the understanding of the strategies use in both our PN and trust models. In this work, we will define the utility of a contract, for an agent, as an aggregation of the weighted utilities of the individual issues as shown below (note this assumes that issues are independent):

$$U^\alpha(O) = \sum_{x \in X(O)} \omega_x \cdot U^\alpha_x(v_x) \tag{4.1}$$

where $\sum \omega_x = 1$ and $v_x \in D_x$ is the value taken by the issue $x \in X(O)$. We consider that agents, whether from the same group or from different groups, invariably interact within some electronic institution (Esteva et al., 2001) which specifies and (or) restricts (some) issue-value assignments of contracts through a set of norms (see section 3.3). An electronic institution, as devised by the system designer, dictates what agents are able to do and say in a given interaction by virtue of their role (e.g. a seller submits asks in an auction while a buyer bids) and the nature of the interaction (e.g. until a winner has been identified, bids are allowed, and then the goods need to be paid for by the winner). Naturally, each institution may also specify different rules.

In the next section, we further describe how we assign weights to different issues in the utility function shown in equation 4.1. The relative weights agents place on each issue of a contract are important in defining what type of game agents play. In particular, we devise these weights such that the agents play the MMPD which extends the usual PD based on our requirements.

4.2 The Multi-Move Prisoner's Dilemma

In defining our trust model and PN model it is intended that they can be implemented in most general applications where agents engage in non-zero sum interactions. In this context, the PD is a common characterisation of interactions between agents that aim to closely model realistic interactions (Axelrod, 1984; Tsebelis, 1990). The PD would, however, limit our models to considering only two types of actions. As per the requirements defined in section 3.3, our trust model needs to be able to adapt to different degrees of reliability of agents (i.e. not be a bistable value), hence different levels of cooperation and defection. Moreover, only defining only two types of actions that can be used as arguments would strongly limit the applicability and efficiency of our PN model in contexts where agents have a large action set to search for an agreement. Given these constraints, we need a *continuous scale* of cooperation between the two extremes that the PD provides us with. To this end, we extend the PD to the MMPD (Prechelt, 1996; Tsebelis, 1990) as shown on figure 4.1. In the MMPD, actions (or moves) are considered to be the enactment of the contents of a contract (e.g. paying for goods, delivering goods). Both the interaction partners have their own actions dictated by the part of the contract that they have to enact (e.g. seller delivers goods and buyer pays for the goods at a given time). Agents may also have more than one issue to take care of (e.g delivery of goods and ensuring they are of a certain quality) and for each issue a discrete number of possible values can be given (e.g. paying after 3 days, 4 days,... or delivering after 1 month, 2 months). When agents engage in the MMPD repeatedly, we term this form of interaction as the Iterated Multi-Move Prisoner's Dilemma (IMMPD). In the following section, we first define the action set (possible moves) of the agents

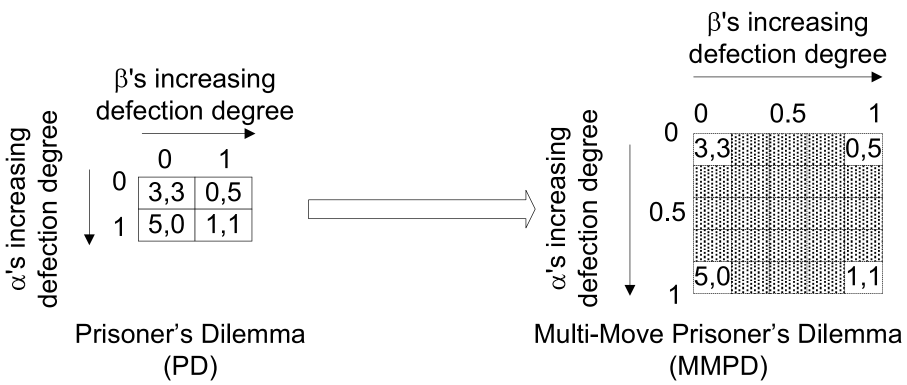


FIGURE 4.1: Transforming the normal Prisoner's Dilemma to the Multi-Move Prisoner's Dilemma. The defection degree increases from 0 to 1 along the direction of the arrows for each agent and the payoffs to each agent is shown in each slot of the game matrix. The shaded region in the MMPD consists of the payoffs of the agents for each degree of defection which we aim to define in terms of the relationship between the utility functions of the agents. Thus, we aim to make the transition from one end of the MMPD to the other a continuous one rather than the discrete one.

which will interact via the MMPD. Then, we provide a formal definition of the MMPD (with respect to multi-issue contracts). The last subsection shows how we can devise the utility functions of the agents so that they can engage in an MMPD. These utility functions are then used by the agents in experiments where we evaluate our trust model and our PN model.

4.2.1 The Action Set

Whenever a contract is signed, each agent is given its part of the contract to enact. In effect, the achievement of the issue-value pairs ($x_i = v_i$) in an agent's part of the contract is its 'action' or 'move' in the game. Achieving what has been agreed or better (for an opponent) is termed cooperation, while doing otherwise is considered a defection. Thus, an agent α can generate its action set $\mathcal{O}(O^\alpha)$ for the MMPD by defining all the possible assignments of the values of the issues that it controls. This is expressed as:

$$\mathcal{O}(O^\alpha) = \left\{O^\alpha = \{x_1 = v_1, ..., x_n = v_n\} \mid x_i \in X(O^\alpha_+), v_i \in D_{x_i}\right\} \tag{4.2}$$

Each agent thus has all its possible actions defined and these actions result in a payoff for each agent similar to a prisoner's dilemma with a discrete multi-action set (as opposed to a binary action set).

4.2.2 The Game

The MMPD is represented as a matrix where each row (and column) corresponds to a particular degree of cooperation from one of the agents (see figure 4.1). Therefore, a contract O between agents α and β can be represented as a point in the matrix where O^α_i is α's action and O^β_k is β's action such that $O = O^\alpha_i \cup O^\beta_k$. The sub-indexes of the different contracts correspond to a row i and a column k respectively in the matrix. We assume that a total order applies over all the possible contracts (in the matrix) according to the utility of each contract to the agent concerned when moving along a single row or column. This means that for an agent α, O^α_i and O^α_j, where $j > i$, are two possible executions but O^α_j is a defection from the agreed contract O resulting in greater utility for α and utility loss for β, if β performs O^β_k (i.e. staying on the same column). Let $\mathcal{O}^\alpha$ be the set of contracts handled by α and $\mathcal{O}^\beta$ similarly for β.

We can then define the multi-move prisoner's dilemma as follows for O^α_j representing a defection from O^α_i by α and O^β_l representing a defection from O^β_k by β:

Definition 4.1. Two agents α and β engage in a Multi-Move Prisoner's Dilemma (MMPD) over the contracts they can execute iff, for any four points in the matrix:

$\forall O^\alpha_i, O^\alpha_j \in \mathcal{O}^\alpha$, where $U^\alpha(O^\alpha_i) < U^\alpha(O^\alpha_j)$ and $\forall O^\beta_k, O^\beta_l \in \mathcal{O}^\beta$ where $U^\beta(O^\beta_k) < U^\beta(O^\beta_l)$, the following rules are respected:

1. Defection Rules (an agent can exploit another's cooperation by defecting but ends

up with a lower payoff if the other side also defects):

$$U^\alpha(O_i^\alpha \cup O_l^\beta) < U^\alpha(O_j^\alpha \cup O_l^\beta) < U^\alpha(O_i^\alpha \cup O_k^\beta) < U^\alpha(O_j^\alpha \cup O_k^\beta),$$
$$U^\beta(O_i^\alpha \cup O_l^\beta) > U^\beta(O_j^\alpha \cup O_l^\beta) > U^\beta(O_i^\alpha \cup O_k^\beta) > U^\beta(O_j^\alpha \cup O_k^\beta),$$

2. Pareto Efficiency Rules (the sum of the rewards when both cooperate is higher than the sum obtained if either or both of the agents defect):

$$U^\alpha(O_i^\alpha \cup O_k^\beta) + U^\beta(O_i^\alpha \cup O_k^\beta) > U^\alpha(O_j^\alpha \cup O_k^\beta) + U^\beta(O_j^\alpha \cup O_k^\beta)$$
$$U^\alpha(O_j^\alpha \cup O_k^\beta) + U^\beta(O_j^\alpha \cup O_k^\beta) > U^\alpha(O_j^\alpha \cup O_l^\beta) + U^\beta(O_j^\alpha \cup O_l^\beta)$$
$$U^\alpha(O_i^\alpha \cup O_l^\beta) + U^\beta(O_i^\alpha \cup O_l^\beta) > U^\alpha(O_j^\alpha \cup O_l^\beta) + U^\beta(O_j^\alpha \cup O_l^\beta)$$
$$U^\alpha(O_i^\alpha \cup O_k^\beta) + U^\beta(O_i^\alpha \cup O_k^\beta) > U^\alpha(O_i^\alpha \cup O_l^\beta) + U^\beta(O_i^\alpha \cup O_l^\beta)$$

From the above rules it is then possible to derive the following payoff matrix for any pair of possible contracts to be enacted by both agents:

		α's part	
		O_i^α	O_j^α
β's part	O_k^β	$U^\beta(O_i^\alpha \cup O_k^\beta), U^\alpha(O_i^\alpha \cup O_k^\beta)$	$U^\beta(O_j^\alpha \cup O_k^\beta), U^\alpha(O_j^\alpha \cup O_k^\beta)$
	O_l^β	$U^\beta(O_i^\alpha \cup O_l^\beta), U^\alpha(O_i^\alpha \cup O_l^\beta)$	$U^\beta(O_j^\alpha \cup O_l^\beta), U^\alpha(O_j^\alpha \cup O_l^\beta)$

TABLE 4.1: The Multi-Move Prisoner's Dilemma

We next define the utility functions that do respect the payoff structure of the MMPD. To this end, we propose the following theorem:

Theorem 4.2. Let X be a given set of issues, α and β be two agents, with X^α being issues under α's control and X^β being issues under β's control (with $X = X^\alpha \cup X^\beta$). Assume that the utility for α of a contract $O = (x_1 = v_1, \ldots, x_n = v_n)$ over issues $X(O) \subseteq X$ is of the form $U^\alpha(O) = \sum_{x_i \in X(O)} \omega_x^\alpha \cdot U_{x_i}^\alpha(v_i)$ and analogously for agent β, $U^\beta(O) = \sum_{x_i \in X(O)} \omega_x^\beta \cdot U_{x_i}^\beta(v_i)$, where $U_{x_i}^\alpha$ and $U_{x_i}^\beta$ are the utility functions for α and β of the individual issue x_i. Moreover we assume that $U_x^\alpha(v)$ and $U_y^\beta(u)$ are differentiable (strictly) increasing functions for any $x \in X^\alpha(O)$ and $y \in X^\beta(O)$ respectively, and differentiable (strictly) decreasing otherwise.

Then, U^α and U^β respect the aforementioned defection and pareto-efficiency rules of a Multi-Move Prisoner's Dilemma (MMPD) if the following conditions are satisfied:

(i)

$$\omega_x^\beta \cdot \left(-\frac{dU_x^\beta}{dx}\right) > \omega_x^\alpha \cdot \frac{dU_x^\alpha}{dx} \tag{4.3}$$

for all issues $x \in X^\alpha(O)$.

(ii)

$$\omega_y^\alpha \cdot (-\frac{dU_y^\alpha}{dy}) > \omega_y^\beta \cdot \frac{dU_y^\beta}{dy} \tag{4.4}$$

for all issues $y \in X^\beta(O)$

where the inequalities are point-wise.

Proof. Without loss of generality, we may assume $X(O) = \{x, y\}, X^\alpha = \{x\}$ and $X^\beta = \{y\}$. Let $O = (x = v, y = u)$ be the agreed contract. We begin by considering a defection by agent α in an issue x from the value v to a value v' such that $U^\alpha(v') > U^\alpha(v)$ (given that everything else remains the same). For an easier notation we will write $U^\alpha(v, u)$ to denote the utility of agent α on a contract $(x = v, y = u)$, similarly for agent β, and $U(v, u)$ for $U^\alpha(v, u) + U^\beta(v, u)$. From the defection and pareto-efficiency rules of the MMPD we have the condition

$$U(v, u) > U(v', u),$$

and using our assumptions on the utilities U^α and U^β (from equations 4.3 and 4.4), this means

$$\omega_x^\alpha U_x^\alpha(v) + \omega_x^\beta U_x^\beta(v) > \omega_x^\alpha U_x^\alpha(v') + \omega_x^\beta U_x^\beta(v') \tag{4.5}$$

That is, we have the equivalent condition to be required:

$$\omega_x^\beta(U_x^\beta(v) - U_x^\beta(v')) > \omega_x^\alpha(U_x^\alpha(v') - U_x^\alpha(v)). \tag{4.6}$$

Now, under general assumptions, we have

$$U_x^\alpha(v') - U_x^\alpha(v) = \int_v^{v'} \frac{dU_x^\alpha}{dx} \cdot dx \tag{4.7}$$

and

$$U_x^\beta(v) - U_x^\beta(v') = -\int_v^{u'} \frac{dU_x^\beta}{dx} \cdot dx, \tag{4.8}$$

Hence, applying the conditions expressed in equations 4.3 of the theorem to equations 4.7 and 4.8 we have equation 4.6 satisfied, and hence $U(v, u) > U(v', u)$ as well (where u' is a defection by α from v). Similarly, the same procedure can be applied to equations 4.7 and 4.8 above using equation 4.4 such that a defection by agent β changing the agreed value $y = u$ to any new value $y = u'$, with $U^\beta(u') > U^\beta(u)$ (given the opponent does not defect in each case), yields $U(v, u) > U(v, u')$.

Finally, if both agents defect to say $x = v'$ and $y = u'$, with $U_x^\alpha(v') > U_x^\alpha(v)$ and $U_y^\beta(u') > U_y^\beta(u)$ (given all else stays the same), then we obviously have the desired inequalities which actually express the pareto-efficiency rules:

$$U(u,v) > \max(U(u,v'), U(u',v)) \geq \min(U(u,v'), U(u',v)) > U(u',v') \tag{4.9}$$

while still having the following defection rules satisfied: $U_x^\alpha(v) < U_x^\alpha(v')$, $U_y^\beta(u) < U_y^\beta(u')$ and $U_y^\alpha(u) > U_y^\alpha(u')$, $U_x^\beta(v) > U_x^\beta(v')$ (given all else stays the same). □

If the utility function of an agent α for each issue in a contract satisfies the conditions expressed in equations 4.3 and 4.4 with respect to its opponent β, then the two agents follow a PD. The transformation of the PD to the MMPD is shown in terms of the game matrix in figure 4.1. As can be see, the binary action set (i.e. cooperation and defection) is transformed into a larger set where agents can enact defections in increasing degrees from 0 to 1.

This characterisation of utility functions is used differently in our persuasive negotiation component and trust model as discussed in the following sections.

4.2.3 Using Persuasive Negotiation in the MMPD

The MMPD can characterise the moves made during negotiations in the form of cooperations and defections. Thus a cooperative move in the MMPD equates to conceding on one's issues (O^α for agent α) in the negotiation, while a defection in the MMPD equates to demanding more on one's issues. If the negotiation mechanism seeks to maximise the social welfare (i.e. the sum of utilities) of the agents in this type of game, each the outcome of the negotiation should be such that each agent concedes more on the issues it likes less and exploits its opponent's less preferred issues. These concessions by the pair of interacting agents has a fixed point in the MMPD, which is known in game theory as the Nash bargaining solution (Osborne and Rubinstein, 1990) (i.e. where the products of their utilities is maximised) as shown on figure 4.2.

However, in repeated negotiation games where agents have different discount factors over repeated games (i.e. the IMMPD), the cooperate-cooperate point in the MMPD may not represent the pareto-efficient point any more. Obviously, in this repeated case the social welfare is higher when agents that have the higher discount factor exploit later games rather than earlier ones while both agents cooperate on the earlier ones. Note that for zero-sum games, the best social welfare is achieved if any of the two agents exploit the first game (and this is likely to be the one with the high discounting effect trying to trade-off the second game in favour of its opponent). This is depicted in figure 4.3. For example, if a buyer highly discounts the value of cars to be bought in future at a low price, the seller (who values future sales more than the buyer) should give a low price to the buyer in the current sale and may increase its price in later sales so that the overall utility is maximised. However, common negotiation techniques neglect

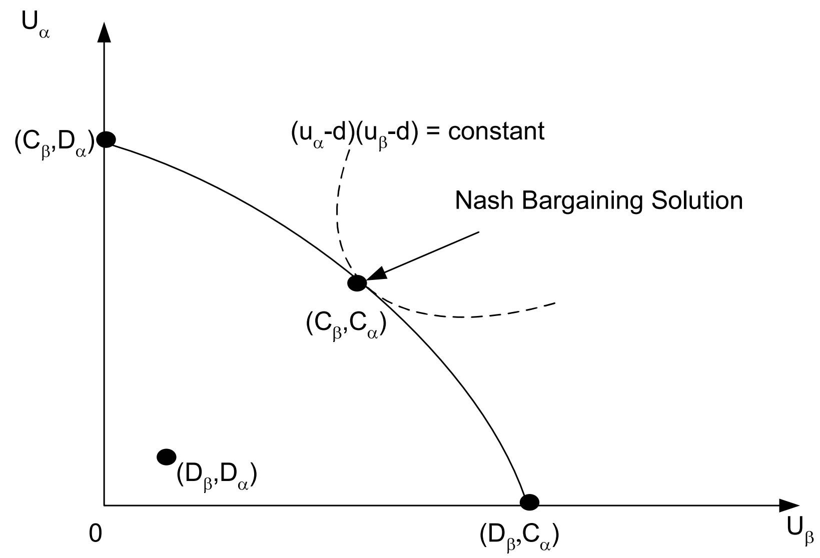


FIGURE 4.2: The social utility (i.e. sum of both agents' utilities) for different negotiation outcomes in the MMPD. C_α means that the agent α cooperates while D_α means that α defects. A higher level of cooperation equates to a higher level of concession in negotiation and a defection equates to demanding more (exploiting the opponent).

this aspect of negotiation and seek only to find the cooperate-cooperate point where no agent is completely exploited in any game (Faratin et al., 1998; Fatima et al., 2004).

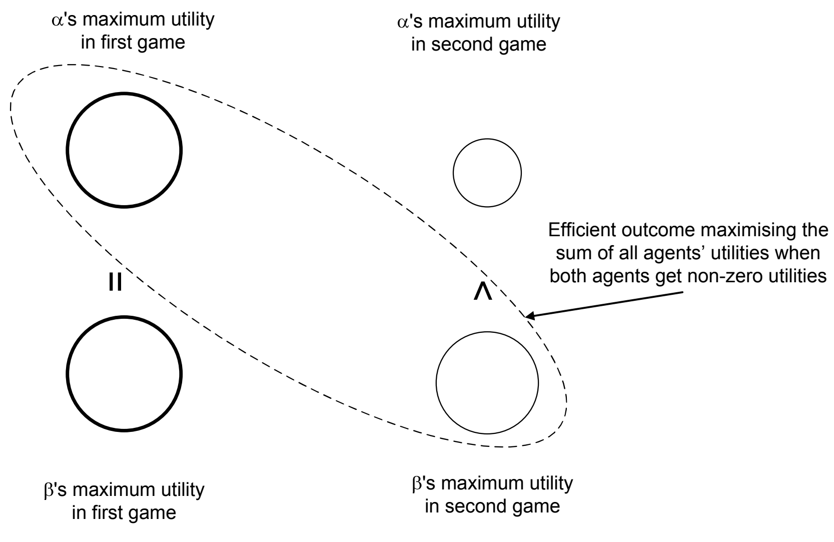
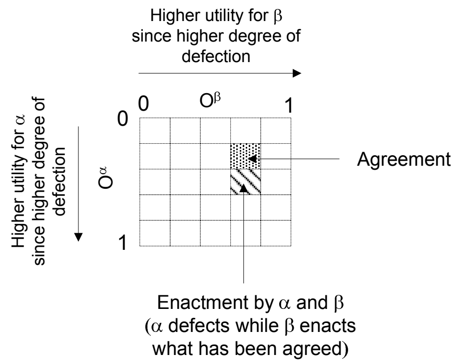


FIGURE 4.3: Choosing the combination of outcomes that maximises the overall utility while ensuring agents have non-zero utilities.

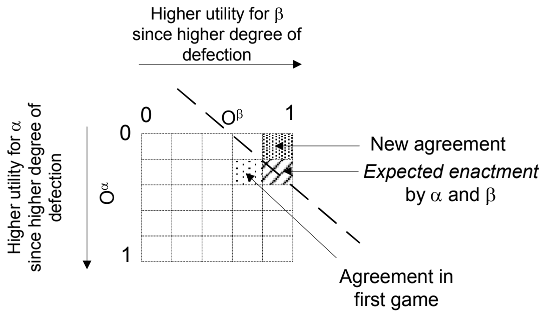
In contrast, through persuasive negotiation and, in particular, the use of rewards, agents may be able to influence the negotiation in such a way that the efficient outcome is reached in this case (i.e. the outcomes circled in dotted line in figure 4.3). Thus, a better social welfare is achieved if agent β which has a high discount factor concedes in earlier games in return for *rewards*, in the form of concessions by α, in the second game as shown on figure 4.3. We elaborate on such a procedure in chapter 7.

4.2.4 Using Trust in the MMPD

A contract is an agreement over the values that issues should take. In the MMPD, this equates to a slot in the game matrix. After an agreement has been signed, agents enact the contents of the contract and may do so with varying degrees of success. Given the structure of the MMPD and the dominant strategy of a PD, the agents will be tempted to defect by enacting values that are more profitable to themselves. This is illustrated on figure 4.4(a) where agent α tries to exploit agent β by enacting the contract in such a way that it obtains a higher utility than what has been agreed in the contract, resulting in a lower utility for β.



(a) α defecting after agreeing with β.



(b) β constraining negotiations (the dotted line) such that agreements are at a higher value and the expected defection still results in an acceptable contract.

FIGURE 4.4: Agents can retaliate using their trust model to capture defections by constraining future agreements.

Agent β may then capture such defections using its trust model and alter its subsequent behaviour accordingly. One of the ways in which this can be achieved is by the agent bounding the range of slots in the MMPD that are available for the next agreement (see dotted line in figure 4.4(b)). This may then reduce the amount of loss expected in the next interaction. As shown in figure 4.4(b), if agent α indeed defects by the same degree, the resulting enactment of the contract results in more utility for β than in the first agreement and this additional utility may compensate to some extent for losses incurred in the first game.

Using the formal definitions in this chapter we can now provide a description of this procedure as follows (the details follow in chapter 5). Assume a trust measure by β's trust model determines the reliability of α to be such that for a given value of $v \in D_x$ for an issue x, α will enact values of x lying in the range $[ev^-, ev^+]$. Therefore, given a negotiation range $[v_{min}, v_{max}]$, an agent may restrict the range to $[v'_{min}, v'_{max}]$ defined as follows:

$$\nu = \{v \mid [ev^-, ev^+] \subseteq [v_{min}, v_{max}]\}$$
$$v'_{min} = \inf\{v \in D_x \mid v \in \nu\}$$
$$v'_{max} = \sup\{v \in D_x \mid v \in \nu\}$$

The above procedure implies restricting the action set of α and β, hence the contract, to only those values where the enactment of the contract can compensate for an expected utility loss due to a possible defection by α. We investigate such a technique as well as other means of adapting an agent's negotiation behaviour in the IMMPD in chapter 5.

4.3 Summary

In this chapter we provided the main formal definitions that are to be used in our trust and PN models. In particular, we defined contracts which describe what agents can offer during negotiation encounters (i.e. the negotiation object) and the utility function that agents can use to evaluate these contracts. Moreover, we provided a description of the MMPD that agents play by virtue of the relationship existing between their utility functions. In so doing, we define the action set of the negotiating agents. This set can be used to define arguments in the PN model and to assess the level of reliability of an opponent by the agents using the trust model. In the latter case, the trust model can subsequently be used to adjust the agent's negotiation stance accordingly.

Given these definitions, in the next chapter we describe our CREDIT trust model. This model allows an agent to determine its level of trust in its opponents based on its own confidence in them and their reputation in the society of agents. Based on CREDIT's trust measure, we further elaborate on the different means that CREDIT provides to adjust the negotiation stance.

Chapter 5

CREDIT: A Trust Model based on Confidence and Reputation

Having reviewed the state of the art models of trust (in chapter 3) and provided the basic definitions in the previous chapter, we now describe our trust model. In particular, we design our trust model as per the general requirements for our negotiation mechanisms detailed in section 1.2 and the particular requirements discussed in relation to other trust models in section 3.3.

To this end, in section 5.1 we first analyse the particular problems that remain with current individual level trust models described in section 3.1 and discuss the techniques we use to solve these problems in our model. Building on this, in section 5.2 we then go on to define the CREDIT trust model. Thus we provide definitions of the context within which agents interact and measures of confidence and reputation that model trust in direct and indirect interactions respectively. Moreover, we provide an algorithm to calculate confidence levels and combine these measures with reputation to generate trust measures. The computational complexity of the algorithm is shown to be linear with respect to the number of past interactions analysed and quadratic with respect to the number of decision variables (here these are fuzzy sets) used to characterise particular levels of reliability. We then show in section 5.3 how CREDIT can be directly used in a bargaining encounter, as per our objectives set in 1.5, to influence agreements reached according to the trustworthiness of the negotiating agents. Section 5.4 empirically evaluates CREDIT and shows that, by influencing the negotiation stance, it is indeed effective and efficient in preventing an agent from being exploited in the long run and in dealing with agents which are reliable to a certain degree. Furthermore, in section 5.5 CREDIT is shown to be better than other comparable models in negotiating fruitful contracts with partially unreliable agents. Section 5.6 summarizes the main properties of CREDIT and discusses the main issues arising in integrating it with bargaining mechanisms and the system-level trust models.

5.1 Introduction

As we have seen in chapter 3, in general, in a society of agents, trust evolves as a result of the direct assessment of the performance of contracted agents over a number of interactions (see section 3.1.1 or from the acquisition of information from the environment, including other agents (see sections 3.1.2 and 3.2.2). More specifically, obtaining a useful measure of trust requires assessing an opponent's performance according to the utility derived from the tasks performed by it (i.e. through *direct interactions*) and public knowledge about its efficiency and effectiveness in each of these tasks (i.e. *indirect interactions*). By considering and combining these two sources (of measures) we believe an agent can better assess the trustworthiness of an opponent, particularly in circumstances where either the private or the public source is not reliable on its own. For example, in cases where an agent has interacted a number of times with its opponent, it will probably rely on[1] When an agent decides to do so will depend on the context within which it finds itself) on its direct measure of its opponent's trustworthiness. However, in cases where the opponent is previously unknown, the agent will rely on the publicly available knowledge. In between, the agent may combine both and give more importance to one or the other depending on the number of interactions it has had with its opponent (Sabater and Sierra, 2002). Nevertheless, whichever measure of trust is used the overall aim is the same; namely to provide an indication of how an agent is likely to perform with respect to a given commitment. Thus, if an agent has been known to defect very often in the past, it may not prove reliable in the future. Similarly, if an agent has frequently proven its effectiveness in past interactions, it may be regarded as reliable in future interactions. Naturally, in some cases, both of these assumptions may turn out to be false.

When measuring trust, it is therefore important to consider the context in which the interactions take place. Here we view this context as being mainly captured by *norms* (Conte and Castelfranchi, 1999; Esteva et al., 2001). Such norms equate to the obligations imposed on the interactions by the system and they occur as a consequence of the utterances, roles, and pledges of the interacting agents. For example, it is the norm in the eBay auction to pay for the goods that one has won before they can be delivered and it is required by mobile service providers (in Britain) to allow seven days to their customers to cancel any contract they may have signed. The relationship between trust and norms is as follows. If it is known that agents act according to certain norms which guarantee good performance, then there is no point in an agent increasing its trust in another that performs well since we cannot assume that it would do so without the norm applying. However, if no norms or rules force agents to behave well (i.e. there exists a possibility to renege), then trust should be increased in an agent which lives up to its

[1] The agent may decide to rely on its own measure of trustworthiness depending on the context. For example, if the agents interact many times in one day and then meet after one month or a year, the direct measure of trust may have become obsolete if the environment is dynamic, while if the environment is static, it may still rely on its own measure after a few days or even a month.

commitments (or decreased if it reneges). Thus, the trust value for a specific agent for a specific task should take into account the potential risk (associated with the task in question) in a contract given information about the norms within which the contract is enacted (Marsh, 1994). This follows from the fact that cooperating under high potential losses (i.e. when no norms enforce good behaviour) shows greater trustworthiness than otherwise (Yamagishi et al., 1998). For example, if no penalty applied for late delivery (i.e. there is no norm regarding delivery times), a seller delivering at the agreed time is deemed more trustworthy than when a harsh penalty applies (i.e. there is a norm that guarantees delivery times).

In general, extant trust models fail to capture all the above-mentioned basic factors in deriving trust (see chapter 3 for more details). In particular, while some of these models devise trust using arbitrary equations that do not take degrees of efficiency into account (see section 3.1.1.2), others simply assume that the trust measure is readily available from the system (see section 3.1.2.1). Moreover, those models which do analyse the performance of opponents fail to consider the norms of the environment which foster good behaviour (López y López et al., 2002; Esteva et al., 2001). Finally, in most applications, extant trust models only use trust in choosing interaction partners and neglect the fact that trust can also be used to adapt the behaviour of an agent towards its opponent at negotiation time (Fisher and Ury, 1983).

Against this background, this chapter develops and evaluates a novel computational trust model (called CREDIT - **C**onfidence and **RE**putation **D**efining **I**nteraction-based **T**rust) that rectifies these shortcomings. Specifically, we show that by taking into account its past experience (from direct interactions) and information gathered from other agents (indirect interactions), an agent can build up beliefs about how trustworthy a contracted agent is likely to be in meeting the expected outcomes of particular contract issues (e.g. delivering goods on time or delivering high quality goods). In this respect, we conceive of two ways of assessing trustworthiness: (i) *Confidence* derived (mainly) from analysing the result of previous interactions with that agent, and (ii) *Reputation* acquired from the experiences of other agents in the community through gossip or by analysing signals sent by an agent. Both measure the same property; that is, the opponent's believed reliability in doing what it says it will regarding particular issues of a contract. In CREDIT both measures rely on (probabilistic) estimations of utility variation in the current contract, based on an agent's past experiences as mentioned and experiences of other agents, which are, in turn, used to evaluate the performance of an opponent by means of a small number of fuzzy sets defining different typical behaviours (in terms of utility variations).[2] The computational complexity of the model is linear with respect to

[2]Fuzzy sets are here used to characterise the vague perception of the performance of an opponent and to provide agents with a high-level of abstraction means of assessing the extent to which an opponent satisfies the issues of a contract. Thus an opponent may be characterised as satisfying with a high degree the (graded/fuzzy) property of 'delivering-on-time' and a with a low degree the (graded/fuzzy) property of 'selling-high-quality', to denote that it is expected to deliver on time and sell goods of relatively poor quality.

the number of past interactions analysed and quadratic in the number of fuzzy sets (in the worst case) in its incremental complexity (i.e. as new interactions occur). Finally, we show how CREDIT can be used in an agent's decision making mechanism in order to minimise risk in interactions by influencing the selection of an interaction partner and the negotiation of contracts.

Set against the requirements of our negotiation mechanisms (which need a technique to model an agent's reliability and honesty in terms of trust) and the challenges that arise in developing such a model (see section 3.3), the work described in this chapter advances the state of the art in the following ways. First, we use the norms of the environment as a key factor in evaluating the trust of opponents. In so doing, we prevent agents from trusting those opponents that are only performing well because of the prevailing norms. Second, we show how fuzzy sets are a very useful tool to describe an agent's probabilistic estimation of its opponent's effectiveness and provides a common ontological basis that permits a combination of this estimation with other agents' estimations. Third, we show how both confidence and reputation measures can be used to develop a measure of trust that is adapted to the environment in which the agents interact and, moreover, how this measure can be adapted over time to become more accurate as more information becomes available. Fourth, we show how CREDIT allows interacting agents, with different norms, to negotiate those issues for which they have different expected values (guided by the norms) and avoid negotiating over those issues for which they have coherent expectations. This, in turn, minimises losses and saves negotiation time. Fifth, we show how trust can be used to adjust the stance that an agent takes during negotiation so as to minimise the utility loss incurred when it believes its opponent is likely to defect by different degrees from a signed contract.

5.2 The CREDIT Model

In this section we define the CREDIT trust model which builds upon our previous work in (Ramchurn et al., 2004d, 2003b). We first provide some new definitions that complement those given in chapter 4 and which we will use in the rest of this chapter. Using these definitions, we model confidence, reputation, and norms. We then show how to combine these measures to compute appropriate trust values according to the environment and the state of an agent. Finally, we analyse the computational complexity of the model.

5.2.1 Rules Dictating Expected Issue-Value Assignments

The agreed contract provides a clear statement of what is expected with respect to each issue. However, the social setting in which the interaction takes place may also give rise to expectations but these are not explicitly stated in the contract itself. For example, a buyer agent α from country A might expect seller agent β from country B to deliver goods nicely wrapped up in gift paper as opposed to in a carton box. This clause may not have been specified in the contract as it is a norm in the client's group that goods must be nicely wrapped up. Thus, at execution time, an agent may fail to satisfy another's (contracted or not) expectations because (i) it is not able to meet the expectations, (ii) it is not willing to meet the expectations, or (iii) it is not aware of the unspecified expectations. In any case, the satisfaction or not of these expectations *directly* impacts on the trust the agent has in its opponent (Molm et al., 2000). If a satisfactory reason is given for poor performance, the trust value may not be modified, but this is not considered here.

Against this background, CREDIT takes into account the three basic sets of norms[3] that can be sources of unspecified expectations[4]: (i) *Social rules*, noted as $SocRules$, that all agents in the society Ag possess in common, (ii) *Group rules*, noted as $GroupRules(G)$, that all agents within a particular group $G \subseteq Ag$ have in common, and (iii) *Institutional rules*, noted as $InstRules$, that agents α and β interacting within a particular electronic institution must abide by. In the case of group rules, there is no guarantee that agents from different groups, having different norms, will satisfy their interaction partner's group rules. On the other hand, the conclusions of institutional rules are guaranteed by the institution (e.g. price c has to be paid, seller has to give goods). This guarantee is normally specified through a penalty which must be paid (by the rule breaker) if the rule is not respected. In more detail, rules of all types allow an agent to infer expected issue-value assignments from a contract. Here the rules will be written in the following way:

$$\textbf{If } x_1 = v_1 \textit{ and } x_2 \geq v_2 \textit{ and ... and } x_m = v_m \textbf{ Then } x \leq v$$

meaning that if $(x_1 = v_1), (x_2 \geq v_2), ..., (x_m = v_m) \in O$, then issue x's value is expected to be equal to v. We assume that x does not appear in the premise of the rule (otherwise this could lead to cyclic rules). An example of such a rule would be:

$$\textbf{IF } \text{price} \geq \pounds 100 \;\; \textit{and} \;\; \text{qos} = 8 \textbf{ Then } \text{anti-DoS} = 10$$

which means that if the price of a telecommunication line (bought from some Internet Service Provider (ISP)) is equal to or greater than a hundred pounds, and the quality of service guarantee (qos) of the ISP is eight (i.e. high in this context), then it is expected that the ISP will provide a very high level (on a scale of 1 to 10 with 10 representing the highest level) of anti denial-of-service (DoS) on the line. We note by $Rules$ the set of all possible rules written using the above syntax[5] over the set X of issues and corresponding

[3]We believe these are the necessary, rather than sufficient, sets of norms that can give rise to unspecified expectations. Other sets of norms could arise from agents creating them or from legal systems for example.

[4]Norms can be of a very complex nature. However, in this paper we *operationalise* norms in the form of constraints that apply over the values of terms in a contract and foresee using richer representations of norms in future work.

[5]Richer syntaxes could also be thought of for premises in these rules, allowing for predicates like

domains of values. The rules an agent abides by will depend on the group it belongs to and the other rules implied by the institution within which it is interacting with others.

Given a contract O proposed by α to β, where $\alpha \in G_1$ and $\beta \in G_2$, we can now devise the set of all of α's (or β's) expectations (unspecified and specified) about the values of the issues in the contract. The unspecified expectations due to the social setting, O^{α}_{exp}, of issue-value assignments from O is the set of all conclusions of the rules of agent α, $Rules(\alpha) = SocRules \cup GroupRules(G_1)$ and $InstRules$ (that apply to α and β), that have their premise satisfied by the equalities in the contract O. The complete expanded contract from α's point of view is therefore defined as $O^{\alpha}_{+} = O \cup O^{\alpha}_{exp}$ (the latter will be different from β's expanded contract, O^{β}_{+}, if β's group have different rules $GroupRules(G_2)$ that apply to the issues of O).

The issues contained in the expanded contract may vary (for the same contract O) depending on the group and institutional rules that apply at the time the agents make an agreement. This is because an agent may interact under different institutions (having different institutional norms) or an agent may decide to switch groups to one that has different norms from its original group. Given the expanded contract, an agent may then decide to trust its opponent depending on its prior knowledge of its opponent's performance. In the next section, we model this in more detail.

5.2.2 Interaction History and Context

In order to try and predict the future performance of an agent it is important to analyse its interaction history in terms of both the outcomes of interactions and the norms that prevailed in each past interaction. In more detail, the interaction history of an agent α, intending to interact with an agent β, can be viewed as consisting of a list of elements with four main components: (i) α's agreed contract O with β and the outcome of the enactment of the contract O' by β and α (i.e. a list of pairs of (O, O') form the contracting history), (ii) $Rules(\alpha)$ that α had to abide by for the contract (at the time t the contract was signed), (iii) $InstRules$ that both α and β had to abide by in a given institution, and (iv) α's utility function (at time t) for the contract issues for which it hired β. Each element in α's interaction history $\Sigma_{\alpha,\beta}$, is therefore represented as:

$$c = \langle \alpha, \beta, O, O', \{U^{\alpha}_{x}\}_{x \in X(O)}, Rules(\alpha), InstRules, t \rangle$$

and the interaction history as $CB = \{c_1, c_2, \ldots\}$. We will note by $CB_{\alpha,\beta} \subseteq CB$, the subset containing all interactions between α and β.

For each new interaction between α and β, α will need to consider the interaction history as well as the currently prevailing rules and its current utility function in order to predict

$>, \geq, <, \leq, \neq$.

the behaviour of β (as will be shown in section 5.2.3.2). Thus we define as α's *current* context[6] within which a new contract is negotiated with an agent β and executed as the set:

$$\Sigma_{\alpha,\beta} = \langle CB_{\alpha,\beta}, \{U_x^\alpha\}_{x\in X}, Rules(\alpha), t_c\rangle. \tag{5.1}$$

where t_c represents the current time. We assume that the agents will have agreed between them (through negotiation or by one partner imposing the institutional rules) which institution will guide their interactions and this will imply a given set of rules $InstRules$ applying over the interaction.[7]

Every time a new contract is agreed and enacted, it is added as a new element to CB in order to update the context of the agent. Moreover, all the rules, including the $InstRules$, will be recorded in the interaction history after the interaction is completed. Thus, this context can be dynamic for a number of reasons (apart from the history being updated with new elements). First, an agent may change groups such that its group rules might change and, consequently, so will its expectations. Second, an agent may interact with the same partners within different institutions (e.g. buying from a seller in England and buying from the same seller in Spain where different trade rules or laws apply). Third, the interacting agents might change their utility functions over time such that they value an issue differently at different points in time (e.g. a travel package may be worth more in summer than in winter).

By taking into account such a dynamic context in evaluating trust, our model can adapt to cases where the environment and the agent are not necessarily static. In the following sections, we use information derived from the context in order to define and evaluate the agent's trust in its opponent's enactment of the contractual terms. We will differentiate between the trust derived from personal knowledge about an agent (confidence) and that derived from information about the agent gathered from other agents in the society (reputation). In the next section we focus on defining confidence (i.e. the personal aspect of trust) and later combine it with reputation (which is based on the confidence of other agents) to get an overall notion of trust.

5.2.3 Confidence

We will define confidence as follows:

α's confidence in β's handling an issue x is a measure of certainty (leading to trust), based on evidence from past direct interactions with β, which allows α to expect a given

[6]Again, we consider these features as necessary rather than sufficient. More features could be added (e.g., social relationships existing between agents or reasons given by an agent explaining its poor performance) and their impact will be investigated in future work.

[7]We do not specify the institutional rules as part of the context since the decision to choose an institution is not defined by the context. However, these rules need to be specified before an agent is able to calculate its trust in its opponent (as will be shown in section 5.2.3.2, that could lead to the choice of the latter as an interaction partner (or not)).

set of utility deviation values to be caused by β's handling of x.[8]

Thus if α has a high degree of confidence with respect to x being well enacted or not by β, then the interval of utility deviation values expected by α from β will be relatively *small* (conversely the set is large if confidence is low). This set of utility deviation values may bring either more utility than expected (i.e. a high confidence in β being 'good') or less utility than expected (i.e. a high confidence in β being 'bad'). We initially consider confidence on a per-issue basis given that agents may be more reliable in satisfying some issues than others. This notion on an opponent's behaviour is not only probabilistic in nature, since it may involve imprecision as well as a subjective appreciation of performance as well (e.g. how 'Bad' or 'Good' the delivery time of goods is for a buyer might not be precisely defined and this perception might also vary over time depending on the agent's preferences). Given this, we choose a fuzzy set based approach to model the meaning of a qualitative term set (e.g. 'bad', 'average', 'good') for performance evaluations in terms of expected utility deviations (and ultimately to expected values for issues), and the confidence level(s) on an opponent refer to the extent to which a particular term fits with her performance. In general, the use of fuzzy sets presents a number of advantages:

1. It allows the modelling of the meaning of imprecise and qualitative terms like 'deliver late' or 'sells high quality goods' which are often used to define the performance of an agent. Using fuzzy sets therefore allows an agent designer to specify the analytical engine of the agent at a higher level of abstraction than using only probabilities.

2. It *does not* require agents to hold the same ontology and objective appreciation of a particular task in order to reliably share information about their opponent's performance, although they may translate such perceptions over a common scale (e.g. utility deviations). For example, not all agents may have the same quality standards for a given product or have the same standards (as a result of different constraints or utility) to judge how late is a delivery by a given seller. Rather, the agents can simply say whether they deem the goods to be 'good' or the seller 'delivers late' to a certain degree and each agent can privately translate this information according to its own notion of 'good' or 'late' into the measure of efficiency all agents use. Thus, agents would only have to share their ontology to understand each other when there is a need to communicate measures rather than define their reasoning mechanism according to exactly the same framework in order to share ratings.

Using this method goes further towards our aim of reducing the uncertainty surrounding

[8] Our definition of *confidence* generally caters for a variety of techniques that could be used to derive confidence values (such as probabilistic measures or time-series analysis). In future work, we aim to define more specific semantics of confidence values and enrich our definition of confidence.

interactions between agents (see section 1.1). In particular, our use of fuzzy sets helps to reduce the uncertainty about the communication mechanisms that agents use to communicate their impressions of others.

5.2.3.1 Confidence Levels

In this work, the behaviour of an agent regarding the fulfillment of an issue in a contract is perceived in terms of the variations on utility between the signed value for the issue and the enacted one. These utility variations are then sensed over multiple interactions to build up a picture of the agent's performance over time. In this thesis we take the stance that fuzzy sets have their domains specified over 'absolute' variations on utility, rather than on relative variations[9] (i.e. relative to the utility of the value signed for the issue). Thus, we consider that $\Delta U \in [-1, 1]$ (recall that utility values belong to the interval $[0, 1]$).

Specifically, we assume that agents share a (small) set $\mathcal{L} = \{L_1, L_2, \ldots, L_k\}$ of linguistic labels to qualify the performance of an agent on each issue. In what follows, we will use the basic set $\mathcal{L} = \{Bad, Average, Good\}$. We believe these labels encompass the whole spectrum of characterisations[10] that an agent might use to express its view on the *possible* (approximate) utility deviations, gains or losses, in the executed contract with respect to the utility of the contractually signed values. For example, each agent could understand the labels 'Bad', 'Average', and 'Good' for the issue 'delivery' in different ways according to their ontology (as shown in table 5.1). As this shows, each agent can have a different ontology to qualify variations between the contracted values and the executed value. We also assume that the translation between the common and the specific terms is private. However, we do require that the common terms have the same agreed upon interpretation among the agents in order to permit a meaningful communication of confidence values among agents (see section 5.2.4).[11] This means that the agents have to share their ontology to perform the translation of terms. Thus, using table 5.1, agent α can translate a 'Very Late' rating from agent β as *Late* (since they both equate to '*Bad*') and 'Right time' from γ as 'On time' (since they both equate to '*Average*'). In more detail, we model the meaning of a label L by a fuzzy set on the domain of utility deviations $\Delta U \in [-1, 1]$, specified by its membership function $\mu_L(\delta u) : [-1, 1] \to [0, 1]$. Examples of membership functions[12] for the above set of labels

[9]This can easily be modified to take into account relative variations depending on the type of opponent encountered and is left as future work.

[10]The sets might be more fine-grained and this will depend on the context of application. The search for the right sets may also be an iterative process where different sets are tried and tested until the ones which fit the goals of the model are found.

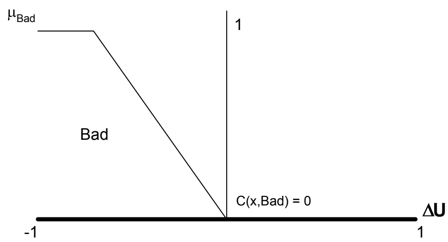
[11]This does not prevent agents from having different perceptions on the variation. Thus some might *perceive* the same variation as significant while others might not. This may happen when agents have different preferences or attribute different weights to the concerned issue.

[12]The shape of the membership function given only serves as an example. Arbitrarily complex functions could be used.

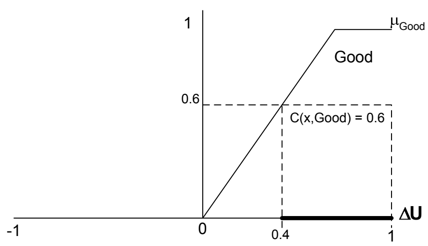
are given in figures 5.1(a), 5.1(b), and 5.1(c).

Label	**Agent**		
	α	β	γ
Bad	Late	Very Late	Too late
Average	On time	Just in time	Right time
Good	Early	Very early	Early enough

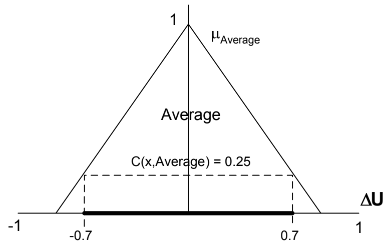
TABLE 5.1: Table showing the possible different meanings of the labels for 3 agents when applied to the issue ‘delivery’.



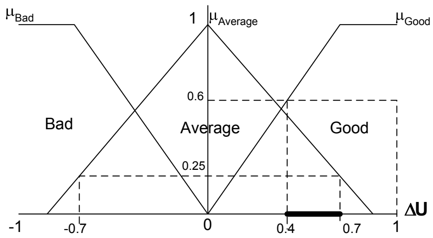
(a) Membership function for label ‘Bad’. The thick line from -1 to 1 represents the support of a confidence level of 0.



(b) Membership function for label ‘Good’. The thick line from 0.4 to 1 represents the support for a confidence level of 0.6.



(c) Membership function for label ‘Average’. The thick line from -0.7 to 0.7 represents the support for a confidence level of 0.25.



(d) Intersection of membership functions. The thick line from 0.4 to 0.7 represents the intersection of the supports for all fuzzy zets given their confidence levels.

FIGURE 5.1: Shapes of membership functions in different labels and ranges supporting confidence levels in ‘Good’ (0.6), ‘Average’ (0.25), and ‘Bad’ (0) as well as the intersection of the supports of these sets.

Thus, agent α’s confidence level is defined as the membership level to a linguistic term L, measured over $[0, 1]$, of the behaviour of a particular agent β, and is noted as $C(\beta, x, L)$. In the rest of this chapter, we will avoid the agent identifier wherever this is unambiguously defined by the context. Therefore, the cut of the fuzzy set L defined by $C(x, L)$ represents a range (on the horizontal axis) of values:

$$E\Delta U_c(x, L) = \{\delta \in [-1, 1] \mid \mu_L(\delta) \geq C(x, L)\} \tag{5.2}$$

that is understood as the range of expected utility deviations at execution time on issue x by agent β. For instance, α may express its belief that β is ‘Good’ to a confidence level 0.6 in fulfilling the contractual values on price, ‘Average’ to a level of 0.25, and ‘Bad’ to a level of 0. This would mean that α expects the utility deviation to lie within the range of values which support the confidence level of 0.6 for 'Good', 0.25 for 'Average', and 0 for 'Bad'. This is shown on figure 5.1.

5.2.3.2 Evaluating Confidence

In order to obtain confidence levels for different labels we first need to calculate the range of utility variations expected for the issue. This expected range can be obtained by considering the utility changes that have been observed in past interactions. There are several techniques to model this range using probability distributions given the size of the sample of ΔU_x that can be obtained from the interaction history (e.g. binomial, normal, or poisson distributions). While the size of the sample of ΔU_x will naturally determine the accuracy of the model, the number of elements in the sample taken (i.e. a window over the latest interactions) will determine how up-to-date the model is in determining the current nature of the opponent. The behaviour of the agent could also be modelled as a time-series so as to predict its behaviour over future time points or analysed using other data-mining techniques (e.g. cluster analysis, neural networks). However, the more complex the analysis, the more time and memory the algorithm will need to devise a level of confidence. Therefore, here we opt for an approximation to a normal distribution[13] which minimises the complexity of calculating the confidence level (see section 5.2.7) and can be tuned to elicit different confidence intervals (e.g. 95%, 99%), for a given sample of ΔU_x.[14]

Using a probability distribution to model utility variations (as opposed to fuzzy sets) does not differ from our initial goal since this probability only models what can be objectively measured and does not take into account the subjective considerations involved in evaluating the range of values obtained (e.g. the extent to which the range of utility variations is detrimental to the agent and the combination of this measure with reputation measures). The subjective perception actually determines which fuzzy set the agent chooses to represent these variations (see section 5.2.3.1).

Therefore, given a context $\Sigma_{\alpha,\beta}$ and a proposed (not yet agreed) contract O, for each issue x in $X(O)$, we can estimate, from the history of past interactions, a probabilistic

[13]The type of probability distribution is not central to the trust model we wish to devise, provided it is continuous and there are techniques to estimate the mean and variance given a small sample of values (since the agent's interactions will certainly not generate the infinite number of samples/points required to model a distribution accurately).

[14]Earlier in section 5.2.3 we highlighted the reasons for using fuzzy sets instead of probabilities. In devising our distribution, we take into consideration all past interactions. This design decision does not permit the model to keep track of a time-dependent behaviour (i.e to cope with time dependent behaviour would simply mean windowing the history to take into account past interactions) and makes the model slow to react to a sudden change in behaviour by the opponent (since the addition of a new element in a large sample may not change its mean or variance significantly). However, it allows us to tune the model to elicit a more precise picture of an opponent when its behaviour does not follow any particular pattern over time and to keep trust sufficiently high for interactions to take place in case the opponent suffers some inefficiencies only for a small number of interactions (i.e. it can forgive bad behaviour).

distribution Φ of α's utility variation $\Delta U_x \in [-1, 1]$ (negative or positive) relative to issue x (we will avoid the agent identifier in the utility function since this is clear from the context). Values of ΔU_x correspond to the possible differences between the utility $U_x^\alpha(v_0)$ of the value $(x = v_0) \in O$ and the utility $U_x^\alpha(v)$ of the (unknown) final value $(x = v)$ in the executed contract O' (i.e. $\Delta U_x = U_x^\alpha(v) - U_x^\alpha(v_0)$). Then we can say that the agent α has a certain *risk* with issue x when it estimates that $q > 0$ where q is the probability that $\Delta U_x < 0$. Of course, the more negative the mean, $\overline{\Delta U_x}$, of this probability distribution (i.e. the higher the expected utility loss), the higher the risk, and the more positive this mean is, the lower the risk (i.e. the lower the expected utility loss).

Thus, to calculate the confidence levels in each of the issues concerned, we first need to estimate the probability distribution of ΔU_x. This has to be done both for those issues x appearing in O and those in the expanded contract $O_+ = O \cup O_{exp}$ resulting from the application of the rules in the current context (see section 5.2.1). We have to do so analogously with the contracts in the precedent cases of the interaction history CB of the current context. However, if we assume that the proposed contract is signed such that the norms of the institution $InstRules$ under which the agents (α and β) are operating are fully enforced (i.e. penalties, matching the utility loss on an issue, have to be paid by the agent which does not respect the norms which regiment the performance on the issue), then the risk is zero[15] for those (groups of) issue-value assignments insured by institutional norms. This is the case even though the inference from previous interactions may suggest that the agent would defect. In such cases, we remove all these insured issues from the analysis. In the same way, if in an element of the interaction history, an issue's enactment was guaranteed by the institution under which the agents interacted at the time, we remove it from the sample of elements being analysed for that issue. This procedure avoids us incrementing trust when an institution has guaranteed good behaviour in the past (since risk is zero in such cases).

Now, assume we have a probability distribution Φ for ΔU_x. In order to determine confidence levels $C(x, L)$ we initially need to determine a significantly representative interval $[\delta_1, \delta_2]$ for ΔU_x (e.g. such that the probability that $(\delta_1 \leq \overline{\Delta U_x} \leq \delta_2)$ is equal to 0.95). This involves first approximating the distribution to a normal distribution by calculating the estimated sample variance $\hat{\sigma}^2$ of the distribution as well as the mean. Then the confidence interval can be obtained from the following equation: $\overline{\Delta U_x} \pm \frac{\hat{\sigma} \cdot l_{conf}}{\sqrt{N}}$, where $l_{conf} = 1.96$ for a 95% confidence and N is the sample size.

Finally, to calculate confidence levels $C(x, L)$ for each label $L \in \mathcal{L}$, we want the interval $[\delta_1, \delta_2]$ to coincide as much as possible with the set of expected values $E\Delta U_c(x, L)$ as computed in equation 5.2. Since this range is defined by the confidence levels of its limits, the procedure amounts to selecting the minimum confidence levels of the two

[15]This assumes that the institution fully insures against any losses. This assumption could be removed and a risk level determined according to the institutional rules as well.

limits for that label as shown in equation 5.3.

$$C(x, L) = \min(\mu_L(\delta_1), \mu_L(\delta_2)) \tag{5.3}$$

We will assume that all agents in the society are able to evaluate their confidence in issues handled by their opponents and may transmit these measures to others. The transmission of such confidence then gives rise to the concept of *reputation* which is described next and later combined with personal confidence measures in section 5.2.5.

5.2.4 Reputation

An agent's reputation is the perception of a group or groups of agents in the society about its abilities and attributes (see section 3.1.2). Several models of reputation have been developed to show how an agent can build up its trust in another by retrieving (see section 3.1.2.1) and aggregating (see section 3.1.2.2) information about the latter from other agents. Thus, here we do not consider how this reputation information is gathered (and aggregated) from the other agents[16] in the society as there already exists several techniques to do this efficiently (Yu and Singh, 2002a; Sabater and Sierra, 2002). Rather, we assume this information is simply available from a social network that structures the knowledge that each agent has of its neighbours and keeps track of past interactions (as per (Sabater and Sierra, 2002)). This allows us to focus on representing reputation and combining it with confidence (as shown in section 5.2.5). In CREDIT, we specialise the definition of reputation to the following:

α's estimate of β's reputation in handling an issue x is α's measure of certainty (leading to trust), based on the aggregation of confidence measures (for x) provided to it by other agents which allows α to expect a given set of values to be achieved by β for x.

These agents may have obtained these confidence values from other agents (i.e. gossiping) or by interacting with β (i.e. witnessing as in (Sabater and Sierra, 2002)). Hence, we assume that an agent α possesses a function $Rep : Ag \times X \times \mathcal{L} \rightarrow [0, 1]$ where $Rep(\beta, x, L)$ represents the reputation of an agent β in handling issue x with respect to the qualifying label L (the name of the agent will be omitted when the context unambiguously determines it).[17] We also assume that the labels $L \in \mathcal{L}$ have their domain specified over the same range of utility deviations (i.e. $\Delta U \in [-1, 1]$) as explained in section 5.2.3.1.

[16] In the case where agent receive conflicting beliefs about its opponent, the agent may choose one of many techniques to tackle this. In REGRET for example, more importance is given to reports from trusted agents while in eBay positive and negative reports are separately reported than aggregated. Depending on how aggregation of reports is performed, an agent may wrongly believe that its opponent is trustworthy. Moreover, in the case where reports do not tally with the behaviour of an opponent, the agent aggregating those reports may less trust the report providers. The problem of conflicting reports are discussed in (Ramchurn et al., 2004b) and a possible solution to this proposed in (Dash et al., 2004).

[17] For more details on such a function, see (Ramchurn et al., 2004d).

In general, reputation measures can be particularly useful to an agent that enters a system for the first time. This is because the agent would not have interacted with any other agents in its environment in the past. Therefore, it would not be possible for it to compute its confidence in them. Thus it can only use information that is supplied to it by other agents in the environment. However, such information may be liable to noise or may not be true if agents are lying. In such circumstances, the agent can only learn from its direct interactions with other agents and compute its confidence measures from these interactions.

As can be seen, using just confidence or just reputation values to compute the set of expected values for a given issue is often only useful in extreme situations. Given this, in the next section we devise a measure that caters for all situations between these extremes and then in section 5.2.6 derive a trust measure from this.

5.2.5 Combined Confidence and Reputation Measures

Generally speaking we consider that both confidence and reputation should be taken into account in order to come up with a set of expected values for an issue. We rely on a combination of both measures in order to balance both the societal view on an opponent and the personal view of the agent until the latter can be sure that its own view is more accurate. We assume in this work that the reputation values expressed by each agent in the society represent their confidence values on the behaviour of a given agent. In other words a value $Rep(x, L)$ represents an aggregation of different confidence values.[18] To come to this conclusion, each agent will have its own threshold on the number of interactions needed to have this accurate measure. Therefore, given agent α's context $\Sigma_{\alpha,\beta} = \langle CB_{\alpha,\beta}, \{U_x^\alpha\}_{x\in X}, Rules(\alpha), t_c\rangle$, here we propose to define the threshold κ as $\kappa = \min(1, |CB_{\alpha,\beta}|/\theta_{min})$, where $|CB_{\alpha,\beta}|$ is the number of interactions of α with β and θ_{min} is the minimum number of interactions (successful negotiations and completed executions[19]) above which only the direct interaction is taken into account (Sabater and Sierra, 2002).

Thus, we capture the combination of confidence and reputation measures through the function $CR : Ag \times X \times \mathcal{L} \to [0, 1]$, which is, in the simplest case, a weighted average of both kinds of degrees (as in the previous cases we omit references to the agent whenever possible and use $CR(x, l)$ instead):

$$CR(x, L) = \kappa \cdot C(x, L) + (1 - \kappa) \cdot Rep(x, L), \tag{5.4}$$

[18]We are therefore implicitly assuming that all these measures are commensurate (i.e have the same meaning and are based on the same scale), and hence their aggregation make sense.

[19]It is important to specify that only those completed interactions should be taken into account since only these can give us information about the behaviour of the opponent in its execution of contracts. Negotiations could end up in no agreements and these should be excluded when counting interactions in the history.

Given CR levels it is then possible to compute the expected values for an issue x and label L as:

$$E\Delta U_{cr}(x, L) = \{\delta \in [-1, 1] \mid \mu_L^x(\delta) \geq CR(x, L)\} \tag{5.5}$$

and then the intersection of the expected ranges for all the labels $L \in \mathcal{L}$:

$$E\Delta U_{cr}(x) = \bigcap_{L \in \mathcal{L}} E\Delta U_{cr}(x, L)\ . \tag{5.6}$$

The assignment of CR values for all labels may not always be consistent (i.e. $E\Delta U_{cr}(x, L) = \emptyset$). This is because each agent in the environment may have undergone different interaction experiences with a particular agent β and each of these agents will obviously transmit different confidence levels for each label. Therefore, in some cases, these confidence levels (when aggregated), may lead to $Rep(x, L_1)$ describing a range of values that does not intersect with that of $Rep(x, L_2)$ or $Rep(x, L_3)$. One solution could be to take the average of intervals to determine a representative interval. However, this may result in a large interval of utility deviations synonymous with a large imprecision in determining the opponent's behaviour. Given this, our straightforward solution to this problem is the following: whenever the intersection results in an empty set, we will iteratively not consider the label with the lowest confidence level, until a non-null range of values is obtained. This procedure equates to removing those decision variables that have the lowest importance in the set under consideration. Our solution ensures that a consistent intersection can be found in all possible cases and minimises the imprecision in modelling an opponent's behaviour given conflicting reports.

As can be seen, the above range is defined in terms of the utility deviations rather than in terms of the values that the issue could take. However, at negotiation time, for example (as will be seen in section 5.3.2.1), we might need to compute the expected values an issue could take, after execution of the contract, given an offered value v_0 for the issue. This requires transferring the expected utility deviations to the domain of the issue considered.[20] This can be computed in the following way:

$$EV_{cr}(x, v_0) = \{v \in D_x \mid U_x(v) - U_x(v_0) \in E\Delta U_{cr}(x)\} \tag{5.7}$$

5.2.6 Trust

In our trust model we use the combined degrees $\{CR(\beta, x, L)\}_{L \in \mathcal{L}}$, as given by equation 5.4, to define the interval of expected values $E\Delta U_{cr}(x)$, that provides us with a maximum expected loss in utility $\Delta_{loss}^{cr}(x) = \inf(E\Delta U_{cr}(x))$ (when $\Delta_{loss}^{cr}(x) < 0$ there is an expected utility loss and when $\Delta_{loss}^{cr}(x) > 0$ there is an expected utility gain). This

[20] By using utility variations, rather than value variations, we can use the same membership functions even if the utility function changes over time.

maximum expected utility loss represents the risk that is involved in the interaction given knowledge acquired both from direct interactions and reputation and also from the norms of the environment. While the risk describes how much we expect to lose at most from an interaction, trust is the opposite of this given our initial definition (see section 5.1). Thus we define α's trust as:

$$T(\beta, x) = \min(1, 1 + \Delta^{cr}_{loss}(x)) \tag{5.8}$$

where T serves to describe α's trust in β for issue x based on both α's confidence in β and β's reputation with respect to issue x.

Here, we choose to bound trust values[21] in the range $[0, 1]$ where 0 represents a completely untrustworthy agent (and corresponds to the maximum possible utility loss) and 1 represents a completely trustworthy agent (and corresponds to zero utility loss).[22]

In any case, we can now define the trust $T(\beta, X(O))$ of an agent α in an agent β over a particular set $X(O) = \{x_1, ..., x_k\}$ of issues appearing in the contract O (or in the expanded one O_+) as an aggregation of the trust in each individual issue (e.g. trust in delivering on time, paying on time and the product having the quality specified in the contract). That is, we postulate:

$$T(\beta, X(O)) = agg(T(\beta, x_1), ..., T(, \beta, x_k)) \tag{5.9}$$

where $agg : [0, 1]^k \rightarrow [0, 1]$ is a *suitable* aggregation function[23]. If some issues are considered to be more important than others, the aggregation function should take this into consideration. This can be achieved by means of different weights[24] w_i given for each issue $x_i \in X(O)$ (the higher the weight, the more important the issue). A typical choice would be to take the aggregation[25] function as a weighted mean:

$$T(\beta, X(O)) = \sum_{x_i \in X(O)} w_i \cdot T(\beta, x_i) \tag{5.10}$$

where $\sum w_i = 1$ and $0 \leq w_i \leq 1$.

[21] We acknowledge that other bounds may be applied in other trust models (e.g. $[-1, 1]$ as in (Marsh, 1994) or $[0, \infty]$ in eBay). See (Marsh, 1994) for a wider discussion on the meaning of the bounds on the rating.

[22] Our choice for the bounds of $[0, 1]$ serves to simplify the analysis when normalising all trust ratings in issues and over contracts.

[23] Generally, an aggregation function is monotonic such that $\min(u_1, ..., u_k) \leq g(u_1, ..., u_k) \leq \max(u_1, ..., u_k)$ (see (Calvo et al., 2002) for a survey).

[24] Most aggregation operators are defined parametrically with respect to weights assigned to each component to be aggregated (see (Calvo et al., 2002) for more details).

[25] More sophisticated aggregation models (based, for example, on different Lebesgue, Choquet, or Sugeno integrals) could also be used (Calvo et al., 2002).

5.2.7 Algorithmic Description and Computational Complexity

Here we detail the algorithm used by CREDIT in generating trust values and analyse its computational complexity. We will assume that reputation information ($Rep(\beta, x, L)$ for all $L \in \mathcal{L}$) about the opponent β has been retrieved by α. Furthermore, we assume that the issues that are guaranteed by the institutional rules, $InstRules$, that apply in the current context $\Sigma_{\alpha,\beta}$ have been removed from the set under consideration (as discussed in section 5.2.3.2). As discussed in section 5.3.2.2, those social and group rules for which their premises have low trust, will introduce more issues to the set under consideration. Finally, we assume that the list of past contracts Ω with β have been retrieved from the interaction history CB.

In table 5.2 we present pseudocode of the function FT used to calculate trust values for a given issue x (according to equation 5.8). As explained in section 5.2.2, the set of precedent cases where an agent $\alpha \in G_1$ has interacted with $\beta \in G_2$, will have been recorded in the interaction history $CB_{\alpha,\beta}$. For each of the elements of $CB_{\alpha,\beta}$, the utility variation is obtained in step 1. Step 2 generates the probability distribution from the list of all utility variations, while step 3 computes the 95% confidence interval of ΔU_x. Step 4a generates the confidence levels ($C(x, L)$ for all $L \in \mathcal{L}$) for each issue using the procedure shown in section 5.2.3.2 while Step 4b combines this measure with reputation to form $CR(x, L)$ for each label. Step 5 simply creates a new instance of all labels to be used in the analysis. In case it is α's first interaction we assume $C(x, L) = 0$ for all $L \in \mathcal{L}$, and start from step 5. Step 6 details the procedure to remove inconsistencies arising out of combining different reputation levels with confidence levels (as was discussed in section 5.2.5, whereby those labels with the lowest confidence levels are removed from the set under consideration). Step 7 checks that inconsistencies have been removed. Step 8 returns the maximum expected utility loss and step 9 returns the trust value using the procedure we later describe in section 5.2.6.

In figure 5.2 we show how the above algorithm fits into the general picture of devising and using trust so as to reduce the uncertainty about the agents' reliability and honesty in interactions and particularly in negotiations (see figure 1.3). As can be seen, steps 1 to 3 implement function g which defines the range of utility variations $[\delta_1, \delta_2]$. Function g could also be implemented using neural networks or cluster analysis in order to elicit the range of utility variations (as discussed in section 5.2.3.2) but here we use a probabilistic approximation to a normal distribution (for reasons specified in section 5.2.3.2). Function f is implemented by step 4. The calculation of trust in steps 5 to 9 eventually results in $T(\alpha, \beta, x)$ and $T(\alpha, \beta)$ while the negotiation range $[v_{min}, v_{max}]$ is specified using the procedure in section 5.3.2.1.

In order to analyse how efficient CREDIT is both in one run of the model and in incremental runs (as more interactions are added to the interaction history), we will derive the computational complexity of each step of the algorithm. This is shown in

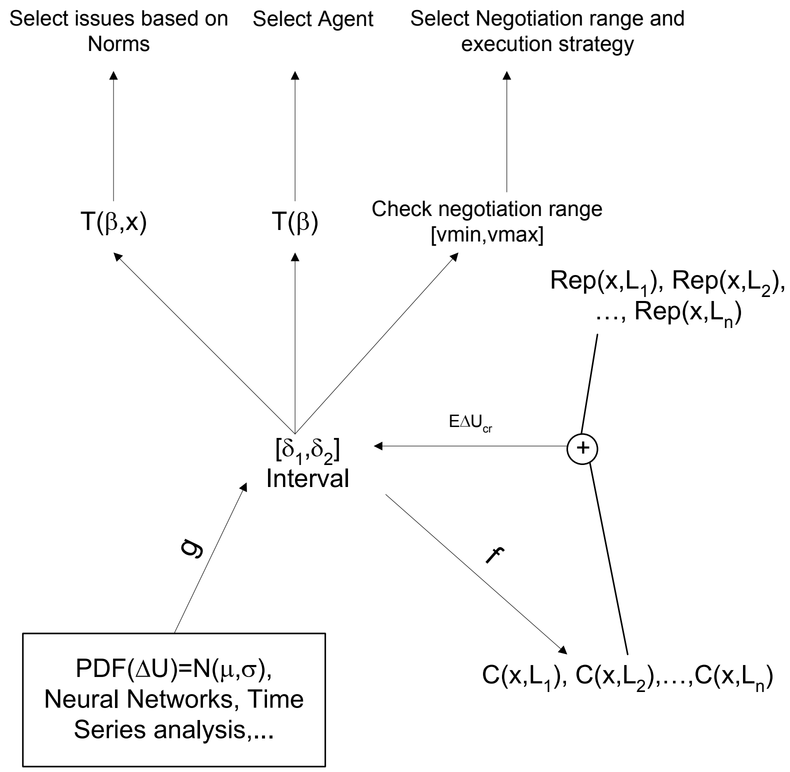


FIGURE 5.2: The processes involved in calculating and using trust with an opponent β. Function g generates the interval given the distribution of utility losses over multiple interactions. Function f evaluates confidence levels as in section 5.2.3.2. Reputation information is assumed to be available.

table 5.3.

As can be seen from table 5.3, the computational complexity of the model when calculating trust on a per issue and per agent basis, is linear with respect to the number of cases in the interaction history when we consider one single run of the model. It is also quadratic with respect to the number of labels (i.e. three in our case - Bad, Average, Good) when evaluated with a new case. This means that, as more and more cases are added (as the agents interact and execute more contracts), the model loops b^2 times in the worst case where b is the number of labels (i.e. when all confidence levels do not coincide with each other).

5.3 CREDIT in Practice

Trust models such as CREDIT can help reduce the uncertainty underlying the honesty and reliability of the interacting agents. However, most trust models are only put to use in selecting interaction partners (see section 3.3). In addition, however, we use CREDIT to influence the bargain that takes place before an agreement is signed (the next chapter is devoted to its application in MD). This is achieved by coupling CREDIT to the decision making model of a agent (see figure 1.3). In this way, CREDIT can directly

`function:` $FT(\beta, x, v, \Omega, U_x^\alpha, \mathcal{L}, \mathcal{M}, R, \kappa)$	
`with`	
$x \in X$;	issue under consideration.
$v \in \mathcal{D}_x$;	the value for issue x.
$\Omega = [(O_1, O'_1), ..., (O_n, O'_n)]$;	the list of past contracts.
$\mathcal{L} = [L_1, ..., L_k], k \geq 2$;	fuzzy sets characterising performance.
$\mathcal{M} = [\mu_{L_1}, ..., \mu_{L_k}]$;	membership functions for fuzzy sets.
$R = \{Rep(x, L)\}_{L \in \mathcal{L}}$;	reputation levels of agent β in all labels.
`begin`	
1. $\Delta U = \{U_x^\alpha(O_i(x)) - U_x^\alpha(O'_i(x))\}_{i=1..n,(O_i,O'_i)\in\Omega}$;	obtaining utility variations in past contract executions.
2. $P \sim N(\overline{\Delta U_x}, \sigma_{\Delta U_x})$;	obtain a Normal probability distribution of utility variations.
3. $\delta_1 = \overline{\Delta U}_x - \frac{\hat{\sigma}_{\Delta U_x} \cdot l_{conf}}{\sqrt{\lvert\Omega\rvert}}$ and $\delta_2 = \overline{\Delta U}_x + \frac{\hat{\sigma}_{\Delta U_x} \cdot l_{conf}}{\sqrt{\lvert\Omega\rvert}}$ with $l_{conf} = 1.96$;	determine the 95% confidence interval of Φ.
4. `for each` $L \in \mathcal{L}$ `do`	
a. $C(\beta, x, L) = \min(\mu_L(\delta_1), \mu_L(\delta_2))$;	obtain confidence levels for each label given the confidence interval.
b. $CR(x, L) = \kappa \cdot C(x, L) + (1 - \kappa) \cdot Rep(x, L)$;	compute combined measure based on confidence and reputation.
5. $\mathcal{L}_{aux} = \mathcal{L}$;	copying labels.
6. `repeat`	
a. $E\Delta U_{cr}(x) = \bigcap_{L \in \mathcal{L}_{aux}} \{u \mid \mu_L^{x=v}(u) \geq CR(x, L)\}$	obtaining range of expected values given reputation and confidence.
b. `if` $(E\Delta U_{cr}(x) = \emptyset)$ `and` $L = \arg\min_{L \in \mathcal{L}_{aux}}\{Rep(x, L)\})$ `then` $\mathcal{L}_{aux} = \mathcal{L}_{aux} - L$;	correct inconsistency by removing low importance sets using linear search (could be logarithmic as well).
7. `until` $E\Delta U_{cr}(x) \neq \emptyset$;	inconsistency removed
8. $\Delta_{loss}^{cr} = \sup(E\Delta U_{cr}(x))$;	calculating maximum possible utility loss .
9. `return` $\min(1, 1 - \Delta_{loss}^{cr})$;	returning the trust value.
`end`	

TABLE 5.2: Algorithm used to calculate trust values.

influence the quality of agreements reached and the efficiency of the negotiation. Thus, in the remainder of this section we examine these two scenarios.

5.3.1 Influencing an Agent's Choice of Interaction Partners

When an agent, say α, has a particular task to contract for, it will decide on the issues to be negotiated and identify possible interaction partners, say $\{\beta_1, \beta_2, ..., \beta_n\} \subseteq Ag$. For each agent in this set, we can calculate the trust value for each issue (as per equation 5.8) and aggregate these to give a general trust value for each agent (using equation 5.9). That is, $T(\alpha, \beta_1, X')$, $T(\alpha, \beta_2, X')$, ..., $T(\alpha, \beta_n, X')$, where $X' \subseteq X$ is the set of issues under consideration. Trust can thus provide an ordering of the agents in terms of their overall reliability for a proposed contract. Agent α can then easily choose the preferred

Step	Complexity	Incremental Complexity
1	$O(n)$	$O(k)$
2	$O(n)$	$O(k)$
3	$O(k)$	$O(k)$
4a	$O(b)$	$O(b)$
4b	$O(b)$	$O(b)$
5	$O(k)$	$O(k)$
6a	$O(b)$	$O(b)$
6b	$O(b)$	$O(b)$
7	$O(k)$	$O(k)$
8	$O(k)$	$O(k)$
9	$O(k)$	$O(k)$
Overall	$O(kn)$	$O(kb^2)$

TABLE 5.3: Computational complexity of individual steps of the algorithm. Here, n is the number of cases recorded in the case base, b is the number of labels and k is a constant.

agent or the set of agents it would want to negotiate with (i.e. by choosing the most trustworthy one(s)).

5.3.2 Influencing an Agent's Negotiation Stance

In the next two subsections we detail ways in which CREDIT can be used to change the negotiation stance. First, we show how CREDIT can be directly used to adapt negotiation intervals on different issues depending on the confidence level for each issue. Second, we show how issues to be negotiated can be varied according to the level of trust in the opponent.

5.3.2.1 Redefining Negotiation Intervals

At contracting time, issue-value assignments, $x_n = v_n$, are agreed upon. Agents accept values that lie within a range $[v_{min}, v_{max}]$, such that $U_x(v_{min}) > 0$ and $U_x(v_{max}) > 0$. This interval is the acceptable range which an agent uses to offer and counter offer (according to a strategy) during negotiation (Jennings et al., 2001). Moreover, given a potential issue-value assignment $x = v$ in an offer, an agent can compute an interval of expected values. Thus, using equation 5.7 we have $EV_{cr}(x, v) = [ev^-, ev^+]$ over which the value v' actually obtained after execution is likely to vary. This range defines the uncertainty in the value of the issue and if the acceptable range $[ev^-, ev^+]$ does not fit within $[v_{min}, v_{max}]$, there exists the possibility that the final value may lie outside the acceptable region. This, in turn, means that $U_x(v')$ may be zero which is clearly undesirable and irrational.

Given this information, there are a number of strategic moves the agent can perform.

First, the agent can restrict the negotiation interval $[v_{min}, v_{max}]$ with respect to the set of expected values $[ev^-, ev^+]$ as shown below. To do this, we first define the set of possible contracts, $\overline{\mathcal{O}}_x$, that are consistent with the expected values of x and its acceptance range, and then define the corrected values for v_{min} and v_{max}:

$$\begin{aligned}
&\overline{\mathcal{O}}_x = \{O \mid (x = v) \in O, EV_{cr}(x, v) \subseteq [v_{min}, v_{max}]\} \\
&v'_{min} = \inf\{v \in D_x \mid (x = v) \in O, O \in \overline{\mathcal{O}}_x\} \\
&v'_{max} = \sup\{v \in D_x \mid (x = v) \in O, O \in \overline{\mathcal{O}}_x\}
\end{aligned} \tag{5.11}$$

This will shrink the range of negotiable values for an issue (i.e. from $[v_{min}, v_{max}]$ to $[v'_{min}, v'_{max}]$, where either $v'_{min} \geq v_{min}$ or $v'_{max} \leq v_{max}$ depending on which of the two limits v'_{min} and v'_{max} gives higher utility respectively) to ensure that the final outcome will fit within the range $[v_{min}, v_{max}]$. As well as reducing the possibility that the executed value will lie outside the acceptable range, reducing the negotiation range can also bring some other added benefits. It can help the agent reduce the time to negotiate over the value of each issue (e.g. if the range is smaller, the number of possible offers is also smaller) and it can help the agent to make better decisions that depend on the negotiation outcome (e.g. if a seller is expected to deliver goods 1 day later than the agreed 3 days, the buyer can adjust its other tasks to fit with delivery in 4 days).

Second, given information about a possible defection on the part of its opponent from an agreed value $x = v_0$, an agent can also decide to defect from its own issues (by a given degree) in order to recover the expected utility loss. This means that the agent will calculate $\min\{U_x(ev^-), U_x(ev^+)\}$ and then achieve $y = u'$ instead of $y = u_0$ (which has been agreed in the contract for issue y which it handles) such that:

$$U_y(u_0) - U_y(u') = \min(0, U_x(v_0) - \min\{U_x(ev^-), U_x(ev^+)\}) \tag{5.12}$$

However, if the opponent is also fitted with a trust model, it will distrust the defecting agent and this may lead to an arms race (Axelrod, 1984; Fisher and Ury, 1983) until the agents will distrust each other so much that they avoid each other (or cannot find a coinciding negotiation range if they both use the procedure described in equation 5.11) .

5.3.2.2 Extending the Set of Negotiable Issues

Initially we argued that higher trust could reduce the negotiation dialogue and lower trust could increase the number of issues negotiated over. In this section we deal with this particular use of trust in defining the issues that need to be negotiated. To this end, issues not explicitly included in a contract O^α may receive an expected value through one of the rules in $Rules(\alpha)$ for an agent α:

$$\mathbf{r} : \textbf{If } x_1 = v_1 \textit{ and } x_2 \geq v_2 \textit{ and ... and } x_m = v_m \textbf{ Then } x \leq v \tag{5.13}$$

Thus, if the premise of such a rule is true in a contract, the issue x is expected to have the value v. If, however, the trust in the agent fulfilling the values of the issues present in the premises is not very high, it means that the agent believes that the values $v_1, v_2, ..., v_n$ may not be eventually satisfied. In such a case, to ensure that the issue x actually receives value v it should be added to the negotiated terms of the contract. This means that, when the trust is low in the premises, the unspecified issues (as discussed in section 5.2.1) are added to the contracted issues in order to try and ensure that they will be met (whereas if trust is high the issue is not negotiated). For example, if a buyer believes that the quality of a product to be delivered (the premise of a rule) will not be the quality of the product actually delivered, the buyer might request that the product satisfies very specific standards (e.g. kitemark or CE), which it privately expected and would not normally specify in a contract if trust were high.

Formally, this means that if $T(\alpha, \beta, X_{\mathbf{r}}) \leq threshold$, (where $(T(\alpha, \beta, X_{\mathbf{r}})$ is defined as per equation 5.9 and $X_{\mathbf{r}}$ is the set of issues in the premise of rule $\mathbf{r}$), then the issue x in the conclusion of the rule should be added to the set of contract terms. On the other hand, as an agent becomes more confident that its interaction partner is actually performing well on the issues in the contract, it might eventually be pointless negotiating on the issue if the premises of the issue pre-suppose that the value expected will actually be obtained. Thus, if $T(\alpha, \beta, X_{\mathbf{r}}) > threshold$, then the issue x in the conclusion of the rule can be removed from the set of contract terms. An example of this would be:

If $T(\alpha, \beta, \text{price}) > 0.8$ and $T(\alpha, \beta, \text{qos}) > 0.7$ **Then** avoid negotiating anti-DoS

which means that the if the trust in provider β giving the quoted price a telecommunication line (bought from some Internet Service Provider (ISP)) is above 0.8, and the trust in its quality of service guarantee (qos) is above 0.7, then the ISP can be trusted to give an anti denial-of-service (DoS) on the line and this issue can be avoided in the negotiation process.

The two processes described above serve to expand and shrink the space of negotiation issues. For a new entrant to the system, for example, the trust value others have in it are likely to be low and hence the number of issues negotiated over will be large. But, as it acquires the trust of others, the number of issues it would need to negotiate will go down. Ultimately, with more trust, the set of negotiable issues can thus be reduced to a minimal set, affording shorter negotiation dialogues. Conversely, with less trust, the negotiable issues expand, trading off the length of dialogues with higher expected utility.

5.4 Evaluating the CREDIT Model

This section empirically evaluates the performance of CREDIT. Here we concentrate on the properties of CREDIT (i.e. in influencing negotiation as shown in section 5.3.2) in two main contexts; namely, in facing normal defectors and those that defect by degrees. We choose these two contexts since these are representative of the behaviours we can reasonably expect agents to exhibit in interactions. Moreover, we use the MMPD to characterise the utility functions as proposed in chapter 4. In this respect, we can describe the three types of agent behaviours as the different ways an agent would enact an agreement as shown in figure 5.3. We build agents that implement such behaviours

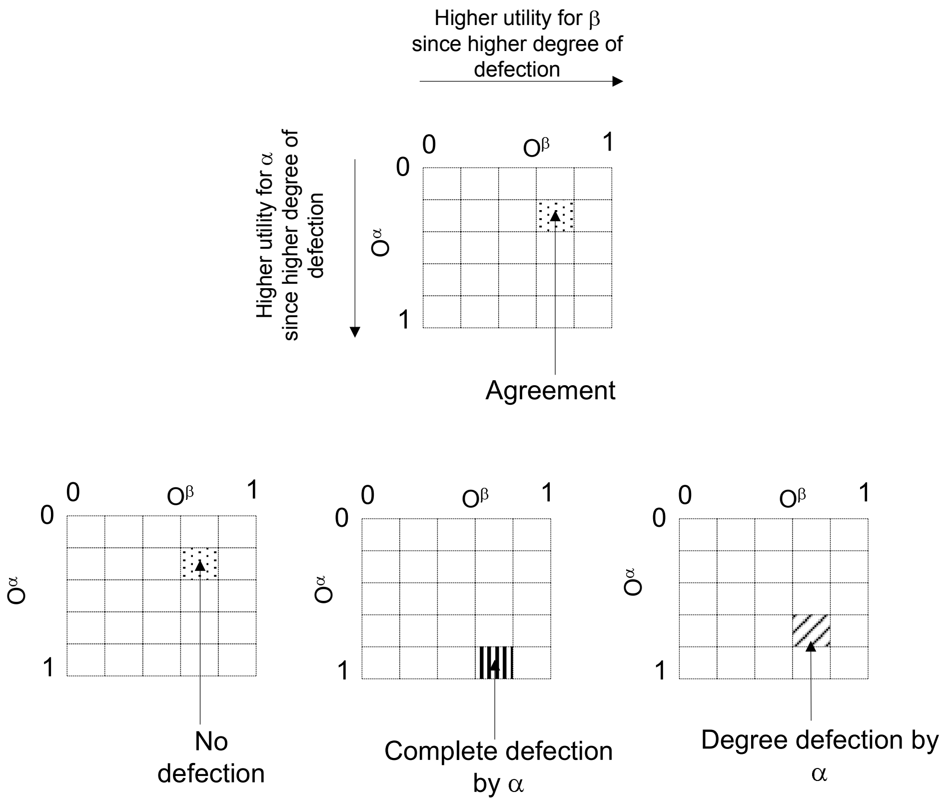


FIGURE 5.3: The different types of behaviours considered in evaluating CREDIT

in their strategy and evaluate CREDIT accordingly. Building on these results, section 5.5 compares the preformance of CREDIT with the main other comparable trust models that have been proposed in the literature.

5.4.1 Bandwidth Trading Scenario

In order to test CREDIT we need a scenario where agents at both ends of the interaction need to trust their interaction partner in negotiating and enacting contracts. These contracts should consist of a number of issues, parts of which are handled by each agent. The agents should also find it profitable to defect on the issues under their control.

While this description fits many scenarios, we specifically wanted to choose one that was *not* developed by us so there is no bias towards the features of our model. As yet, there is no common benchmark test in this area and so we chose the scenario of (Witkowski et al., 2001). The latter model has the advantage of providing *both* agents (e.g. the buyer and the seller) involved in an interaction to defect for a number of reasons which we elaborate upon and augment. Here, a number of service provider agents are interacting with a number of users buying telecommunication bandwidth over a given channel. As an example, the service providers could be Internet Service Providers (ISPs) selling connection on an Asymmetric Digital Subscriber Line (ADSL) link to individual home users who use the link to browse the internet or download movies. Thus, service provider agents (SP) provide bandwidth for telecommunication for a price $c \in [30, 100]$, and of size $s \in [2Mbits/s, 4Mbits/s]$. Service users (SU) buy the service through a negotiated agreement to pay for the service at time t_c (normally between 10 days and 30 days) after the contract is made. The agents also negotiate the security level, $l \in [1, 6]$ (where $l = 1$ is the lowest security level and $l = 6$ is the highest security level), that the channel users will respect (e.g. a channel of $l = 6$ will prevent denial of service attacks or use a firewall to prevent worms infiltrating, but might also block common user programs such as instant messengers). SU agents will negotiate for lower security levels so as to be able to use the bandwidth for various types of programs, while SP agents will negotiate for higher security requirements to keep its services robust against attacks. We assume that users and providers belong to different groups (e.g. SU can be Academic users, Home users and Business users, while SP can be International ISPs, Local ISPs, Home ISPs) and therefore have different social rules and norms but interact within the same institution (e.g. the British trade laws).

5.4.1.1 Specified and Unspecified Issues

Generally speaking, agents agree on the price, size of the bandwidth, the time of payment, and security level. Other issues that may get included in a contract include (i) the quality of service level, $qos \in [1, 8]$ (where $qos = 1$ is the lowest quality and $qos = 8$ is the highest quality, each specifying error rates or download speed for example), and (ii) the connection usage of the SU agent, $usage \in [5, 100]$ (i.e. in terms of the number of http requests per minute). The addition of such issues will depend on the social rules and norms that pervade the interaction.

While the SP agents might believe that a size and price of bandwidth equates to a certain level of qos (i.e. through a rule **If** $c \leq 35$ *and* $s = 4Mbits/s$ **Then** $qos \leq 6$), the SU agents might believe other values for the qos (i.e. **If** $c \geq 32$ *and* $s = 4Mbits/s$ **Then** $qos \geq 4$). Moreover, SU agents may have different norms regarding the expected bandwidth usage of the service given the time of payment and security level agreed for the bandwidth (i.e. **If** $l \leq 5$ *and* $t_c = 15$ **Then** $usage \geq 60$) as compared to the SP agent (i.e. **If** $l \geq 5$ *and* $t_c = 15$ **Then** $usage \leq 75$). Thus, if an agent cannot be trusted on the issues (it handles) that form the premises of such rules, the issue involved in the conclusion of the rule gets added to the contract as shown in section 5.3.2.2. Finally, all agents have the same institutional rule concerning the method of payment; *cash* if below a certain value and *credit card* if above a certain value (i.e. **If** $p > 50$ **Then** *creditcard* and **IF** $p < 50$ **Then** *cash*). This means that the institution guarantees that the payments will be made by the appropriate method, otherwise penalties are paid by the defector such that the other agent loses nothing (e.g. using a credit card to pay for very low amounts of money causes an additional commission to be charged by the credit card company). Adding new issues to the negotiation equates to expanding the negotiation object (see section 1.2). In terms of the MMPD which agents are playing, this procedure equates to scaling up the game matrix to consider a larger space of possible degrees of defections from a given agreement. This is because an additional issue in a contract increases the range of possible enactments of a contract and therefore increases the range of possible defections. The rules are summarised in tables 5.4 and 5.5.

Agent	GroupRules 1	GroupRules 2
SP	**If** $p \leq 35$ *and* $s = 4Mbits/s$ **Then** $qos \leq 6$	**If** $l \geq 5$ *and* $t_c = 15$ **Then** $usage \leq 75$
SU	**If** $p \geq 35$ *and* $s = 4Mbits/s$ **Then** $qos \geq 4$	**If** $l \leq 5$ *and* $t_c = 15$ **Then** $usage \geq 60$

TABLE 5.4: Group Rules that apply to quality of service and usage.

Agent	Institutional Rule 1	Institutional Rule 2
SP and SU	**IF** $p < 50$ **Then** *cash*	**IF** $p \geq 50$ **Then** *creditcard*

TABLE 5.5: Institutional rules that apply to the payment method.

5.4.1.2 Defections and Cooperation

The opportunities for defection or cooperation are identified for both agents regarding the main issues as shown below. Here we assume that β is the SP agent and α is the SU agent.

1. Specified Issues (forming O constituting of issues that are always negotiated upon, as explained in section 5.2.1).

   (a) Price c – SP agents can defect after reaching an agreement by subsequently asking for a higher price than c or cooperate by asking for an equal (or lower) price than c. In the MMPD, this means that $p \in X(O^{\beta})$ and a higher price demanded equates to a higher level of defection.

   (b) Time of payment t_c – SU agents can defect by paying later than t_c or cooperate by paying at time t_c or earlier (paying later allows them to manage their funds better – hence higher utility). In the MMPD, this means that $t_c \in X(O^{\alpha})$ and a later price means a higher degree of defection by α.

(c) Size s – SP agents can defect by providing a lower bandwidth (using the bandwidth elsewhere brings them higher utility) or cooperate by providing the required size or more. Depending on the institution, the size may or may not be changed at execution time. In the MMPD, this means that $s \in X(O^{\beta})$ and a lower bandwidth equates to a higher level of defection.

(d) Security level l – SU agents can defect on this issue by using the bandwidth to transmit spam and viruses in order to gain some economic benefit by sending mass advertising email or to attack other users and damage their system so as to get a larger share of the market they might be trading in. In the MMPD, this means that $l \in X(O^{\alpha})$ and abusing the bandwidth means a higher degree of defection by α.

2. Optional Issues (forming O_{exp} constituting of those issues that are negotiated only if trust is low in the specified issues above as explained in sections 5.2.1 and 5.3.2.2).

   (a) Quality of service, *qos* - the *qos* is only added to the set of negotiable issues by an SU agent if the SP agent it wishes to contract cannot be trusted on the price and the size to be provided (see table 5.4). If the *qos* is added to the contract, an SP agent could defect from the agreed value since providing a service of low quality results in a higher payoff for the SP agent and lower payoffs for the user agents. In the MMPD, adding this issue equates to extending the game matrix (shown in figure 5.3) horizontally since this issue provides β with a larger range of issues to defect on.

   (b) Connection usage, *usage* - the *usage* is only added to the set of negotiable issues by an SP agent if the SU agent wishing to contract it cannot be trusted on the time of payment and the security restrictions it should normally respect (see table 5.4). SU agents can additionally defect from an agreed *usage* by loading the bandwidth more than agreed causing a loss in efficiency in the SP's servers managing the channel. In the MMPD, adding this issue equates to extending the game matrix (shown in figure 5.3) vertically since this issue provides α with a larger range of issues to defect on.

Prior to contract execution, the agents engage in negotiations to reach an agreement over the above-named issues. Once a contract is signed, the agents commit themselves to the values agreed upon. Different values for the above issues result in different executions of the contract, each with a different utility to both agents. Defections result from achieving the values which give a utility gain to one agent (SP or SU) and a utility loss to the other (SU or SP). A walk through the use of CREDIT, taking into consideration all the above factors, is given in appendix B. The experimental setup is described in the following section.

5.4.2 Experimental Setup

There are two aspects of CREDIT to be tested (as discussed in section 5.3): (i) defining issues to be negotiated upon; and (ii) defining negotiation intervals. As was pointed out in section 5.3, CREDIT also performs partner selection based on trust. However, the behaviour of the model in this respect follows from the results of the other two mentioned above since these will also show how well CREDIT detects defectors and how fast the trust value changes accordingly. Therefore, to evaluate whether the CREDIT model does indeed bring added value to the agents, it is necessary to show how agents using the model can identify and cope with agents of different *execution* strategies[26] with respect to enacting the contents of a contract.

In more detail, we will show how the trust model helps agents cope with other agents which (i) either defect (i.e. achieve the worst possible values for issues for their opponent) or cooperate (i.e. enact the contract) completely, and (ii) defect in degrees (i.e defecting from the values agreed to a limited extent). These execution strategies exploit the basic moves we described in the MMPD in figure 5.3 (see beginning of section 5.4). To this end we devise two sets of experiments with execution strategies as defined below[27]:

1. Experimental set 1 deals with *extreme* defection or cooperative execution strategies (i.e. with maximum defection or cooperation degrees or both):
   (a) philanthropic (P) – never defects, and always delivers what has been agreed in the contract.
   (b) nasty (N) – always defects maximally and achieves whatever brings it maximum expected utility.
   (c) tit-for-tat (TFT) – defects when the opponent defects but cooperates fully on the first contract.
   (d) strategic defector ($STDefect$) – defects and cooperates alternatively in an attempt to keep up its opponent's trust on it, thus exploiting the latter.
2. Experimental set 2 deals with agents that have fixed degrees of defection. To this end we define the degree defector (DD_d) as a defector which defects from an agreed value by a degree d in the range $d \in [0, 1]$. The value d represents the maximum fraction (of the range of values that the issue can take) that the agent will defect by.

These two experimental sets generally cater for behaviours commonly encountered in e-commerce. For example, degree defectors could represent inefficient companies, com-

[26] Here we distinguish an execution strategy as being the behaviour an agent adopts when enacting the contents of a contract from a negotiation strategy that is used in negotiating the contract.

[27] We exploit those strategies commonly used in assessing trust models (since Axelrod's experiments (Axelrod, 1984)) but we modify them for our context.

plete defectors represent hackers or spammers while philanthropic and tit-for-tat agents represent well established companies or sellers (e.g. on eBay or Amazon).

In the remainder of the chapter, we will denote the strategy used by an agent by tagging the strategy identifier with the role of the agent. For example, P-SP denotes a philanthropic SP agent and N-SU denotes a nasty SU agent. Wherever we will need to test our results for statistical significance we will use ANOVA (ANalysis Of VAriance between groups) to analyse the means of samples of different sizes to ensure that our means indeed exhibit the properties we seek, and as a result to prove or disprove our hypotheses.

The same fuzzy sets applying over utility deviations are given to each agent to characterise the performance of an opponent's issues of a contract.[28] Specifically, the three fuzzy sets Bad, $Medium$, and $Good$ are defined using linear functions based on figure 5.1. The basic settings for these experiments are summarised in table 5.6 and the utility functions together with the weight each issue has in the overall utility of each type of agent are given in table 5.7.

θ_{min}	50
No. of specified issues	4
No. of unspecified issues	2
Institution	Rules defined as per table 5.5
Fuzzy Sets	Bad, Average, Good
Level of Confidence of Risk	95%

TABLE 5.6: Basic settings of the experiments for sets 1, 2 and 3.

	U_c, ω_c	U_s, ω_s	U_l, ω_l	U_{t_c}, ω_{t_c}	U_{qos}, ω_{qos}	$U_{usage}, \omega_{usage}$
SP	$\frac{c}{200}, 0.05$	$1-\frac{s}{12}, 0.05$	$\frac{l}{6}, 0.1$	$1-t_c, 0.1$	$1-\frac{q}{16}, 0.5$	$1-\frac{usage}{100}, 0.2$
SU	$1-\frac{c}{100}, 0.5$	$\frac{s}{4}, 0.2$	$1-\frac{l}{10}, 0.1$	$\frac{t_c}{90}, 0.1$	$\frac{q}{8}, 0.05$	$\frac{usage}{200}, 0.05$

TABLE 5.7: Utility functions used in the experiments for SP and SU agents. Note that all agents of each type (SU or SP) have the same utility functions but may have different execution strategies.

These weights were chosen such that the agents play the MMPD (see section 4.2). Moreover, more weight is given to 'specified' issues than to 'unspecified' ones given that we expect an agent to consider those specified issues as more important than the unspecified ones (since the former are *always* negotiated). Finally, the calculation of the overall trust value for each type of agent is given in table 5.8. As can be seen, the SU agent weighs its trust in each issue respecting the order of the weight each issue has in

[28]We expect these fuzzy sets to be different for each agent in realistic applications. This difference in perception (which fuzzy sets express) will matter whenever agents are meant to exchange reputation values. However, this is a feature which we do not use here since the reputation values are assumed to be available from the society and we wish to keep a focus on the analysis of direct interactions rather than delve into the topic of aggregation of reputation values. See (Ramchurn et al., 2004b) for a wider discussion on reputation and trust.

its utility function (and similarly for the SP agent). This assumes the agent will choose an opponent it trusts most on those issues it considers most important. In remaining

$T(\alpha, \beta) = 0.5 \times T(\alpha, \beta, c) + 0.4 \times T(\alpha, \beta, s) + 0.1 \times T(\alpha, \beta, qos)$
$T(\beta, \alpha) = 0.5 \times T(\beta, \alpha, t_c) + 0.4 \times T(\beta, \alpha, l) + 0.1 \times T(\beta, \alpha, usage)$

TABLE 5.8: Calculation of trust values for an SU agent α and an SP agent β.

subsections we detail the different experiments performed to test CREDIT when used by agents with different strategies. The behaviour of CREDIT is not specifically tested for one shot interactions. In such cases, we expect CREDIT to use the reputation model connected to it (e.g. REGRET, SPORAS or HISTOS) to dictate what should the behaviour be (Ramchurn et al., 2004d). Moreover, in the one-shot interaction case where the only interaction partner available has a low reputation, an agent might choose to interact within the framework of an institution which guarantees all or most of the terms of contracts they make.

5.4.3 Experimental Set 1: Facing Extreme Strategies

In this set of experiments, pairs of agents with extreme execution strategies 1 to 4 (as per section 5.4.2) negotiate contracts with each other and enact them after coming to an agreement.

5.4.3.1 Using Norms and Trust in Negotiation

Having proposed to use a combination of norms and trust at negotiation time in section 5.3.2.2, here we test CREDIT to see whether this combination can actually enhance negotiation encounters. Specifically, we applied the following rules to the issues (based on those rules explained in section 5.3.2.2):

- **Rule 1** : **If** $T(SU, SP, c) \geq 0.9$ *and* $T(SU, SP, s) > 0.95$ **Then** avoid negotiating qos.[29]

  This rule means that the SU agent will avoid negotiating the quality of service if it trusts that the SP agent will not defect on the price and the size of bandwidth. This is based on a norm in SU's group which says that *qos* is normally understood to be of a given type if the price and size of bandwith are of a certain value (see table 5.4). The same norm might not apply in the SP's group.

[29] We use thresholds above or equal to 0.9 to indicate a high threshold on trust. This imposes stringent conditions on the trustworthiness of the opponent for the issues the proponent values most. Lower thresholds could be imposed to compensate for any noise in perceiving the performance of the opponent, but we do not consider this here.

- **Rule 2** : **If** $T(SP, SU, t_c) \geq 0.99$ *and* $T(SP, SU, l) > 0.9$ **Then** avoid negotiating *usage*.

  This rule means that the SP agent will avoid negotiating bandwidth usage if it trusts the SU agent will honour its payment in time and if it satisfies the security level (see table 5.4).

Given our expectations regarding the effect of norms and trust on the negotiation process and in particular on the number of offers exchanged in the process (i.e. the negotiation thread) which may determine the time taken to come to an agreement, we postulate the following hypothesis:

Hypothesis 1. The more issues negotiated due to trust being low, the lengthier will be the negotiation thread using standard negotiation strategies.

In order to test this hypothesis, we set a P-SU agent to negotiate with a P-SP agent given the rules set above.[30] For this experiment, the two agents use off-the-shelf negotiation strategies such as Relative and Absolute tit-for-tat.[31] In order to vary the number of issues to be considered at negotiation time, we reduced each agent's trust in the premises of the norms accordingly and then kept the trust values fixed for each subset of these experiments. For example, one subset of the experiments would involve only **Rule 1** above not firing given that trust in price or size would be high (i.e. $T(SU, SP, c) > 0.99$). In order to keep all other variables constant, CREDIT was prevented from modifying the negotiation ranges at runtime as well (as per section 5.3.2.1) since changing negotiation ranges changes the number of possible agreements and the value of those agreements to the agents.

	Rule 2	
Rule 1	**Fires**	**Does not Fire**
Fires	3.084	3.286
Does not Fire	3.308	3.47

TABLE 5.9: The effect of norms on the average length of the negotiation thread needed to reach an agreement (results from P-SU v/s P-SP with rules 1 and 2 firing alternatively and together)

Table 5.9 outlines the effect of these rules on the negotiation encounters. As can be seen, our hypothesis is validated since agents can reduce the length of negotiation threads (by 11% in the best case) needed on average to reach an agreement whenever they trust

[30] In this experiment, we disconnect the trust evaluation (used in changing negotiation ranges) component of CREDIT in order to specify conditions where trust is low independent of the strategy and thus simplify the analysis we wish to make here.

[31] Other negotiation strategies could be used but the variable we study here is not strictly dependent on the negotiation strategy and it is only our intention to show that a difference in the number of issues considered will affect the negotiation efficiency. We therefore use those simple negotiation strategies that rely on very few heuristics for ease of analysis.

No. of issues	4	5a	5b	6
Average Utility	0.465	0.4640	0.4645	0.4647

Table 5.10: The effect of the number of issues negotiated on the average utility of agreements for a P-SU agent (results from P-SU v/s P-SP). 5a and 5b represent the five issues that are negotiated given that only one rule fires for either the SP or SU respectively.

their counterparts.[32] While this enables agents to achieve agreements faster (i.e. they might take less time to negotiate), it is questionable whether the conclusion of the rules (i.e. *qos* and *usage*) should be negotiated at all.[33] This is because, even though these issues are given acceptable values according to the norms, it cannot be guaranteed that these values are the best that could be achieved given the preferences of both agents negotiating. In fact, we might expect a trade-off between accelerating negotiation based on norms and making better agreements (i.e. achieving higher utility). However, from the above experiments it was also found that the gain (or loss) in average utility achieved was **not** substantial when more issues were negotiated as can be seen in table 5.10 (in this case the agent lost utility when more issues were negotiated since the newly negotiated issues are assigned lower values than those they usually get when trust is high and the issue is not negotiated).[34] This is because the issues, only negotiated due to rules not firing, do not have substantial weight in the utility function of the agent. Otherwise, we would expect these issues to form part of the initial negotiable set (e.g. price and size are important issues that need to be negotiated since they contribute significantly to the utility of the agent).

In summary, the above results tell us that CREDIT will cause fewer issues to be negotiated when trust is high and more issues to be negotiated when trust is low. Moreover, CREDIT has been shown not to significantly reduce the maximum achievable utility in negotiated contracts by applying norms (when the issues added do not significantly impact on the utility of agents). Therefore, we decided to keep the rules above in future negotiations in the following experiments in order to speed up negotiations.

[32]These results were tested for statistical significance using ANOVA (single factor) and the null hypothesis (i.e. that the means of the groups are the same) was invalidated. This follows from the fact that $F = 9.07 > F_{crit} = 3.5 > 1$, with $p = 5.8 \times 10^{-6}$,$\alpha = 0.01$, and a sample size of 500. This result proves that each rule indeed has an effect on the outcome. To further investigate the interaction between different rules, we used ANOVA on the results for rule 1 and rule 2 alone firing. The results are as follows for $\alpha = 0.025$ and a sample size of 500: $F = 5.004 < F_{crit} = 5.006$, and $p = 0.0258 > \alpha$. As can be seen, the means do not significantly differ in this context (i.e. since $F < F_{crit}$). This is because these two samples actually test the negotiation length using the same number of issues where these issues have been obtained from different rules firing. The low value of $m = 3.084$ can be explained by the fact that less issues have to be negotiated (i.e. four issues as compared to five or six in the other cases).

[33]In our experiments, when the issues are not negotiated, we choose the value lying in the middle of the intersection of the agents' acceptable ranges for these issues.

[34]These results were checked for statistical significance using ANOVA (single factor) which tries to identify significant differences between means of different samples (of 500 elements) of utility of contracts. Thus, the means obtained were found not to be significantly different (i.e. the null hypothesis that the means are the same is validated) given $F = 0.08 < F_{crit} = 3.6$ with $\alpha = 0.12$. This means that the difference in means is more due to chance than the number of issues' influence.

5.4.3.2 Trust and Negotiation Intervals

Here, we aim to assess how well CREDIT can recognize defectors and adjust its negotiation intervals accordingly. We therefore tested CREDIT on pairs of agents with each combination of strategies (we consider all strategies except degree defectors) in sequence and recorded the number of contracts agreed upon, the trust values throughout the experiments, the utility of contracts achieved and executed, and the number of issues negotiated.[35]

There are four experimental variables that we will vary in order to see their impact on the behaviour of the agents (note that agents cannot change their trust value during negotiations): (i) the execution (as opposed to the negotiation) strategies used by pairs of agents, (ii) the reputation of the interacting agents, (iii) the point at which the agreement is made, and (iv) the extent to which the negotiation ranges of pairs of agents coincide. The first two variables can be preset by simply pairing agents with different (or same) execution strategies (e.g. TFT v/s N or P v/s P) and hardwiring the reputation levels on each fuzzy set at different levels (and elicit different initial trust values). The agreement point could be set by the agents' negotiation strategies. However, using such strategies results in a different agreement point for each negotiation such that it is difficult to extract the general trend and analyse it. Therefore we set the agreeement points as follows.

From figure 5.4, we can set the degree of alignment between negotiation ranges, λ, as:

$$\lambda = \frac{w}{z} \tag{5.14}$$

where z is the whole range covered by the negotiation ranges and w is the range of values describing the intersection of the negotiation ranges.

Given that the negotiation range of agent α is noted as $[v^{\alpha}_{min}, v^{\alpha}_{max}]$, and that of agent β is noted as $[v^{\beta}_{min}, v^{\beta}_{max}]$, and assuming that v^{α}_{min} and v^{β}_{max} are fixed, v^{α}_{max} and v^{β}_{min} can easily be adjusted to give different degrees of alignment (i.e. by changing w).

On the other hand, the negotiation power[36] of agent α, ϕ is set as follows:

$$\phi = 1 - \frac{w'}{w} \tag{5.15}$$

where w' is the distance of the agreement from the lower bound of the intersection (here the lower bound is of higher utility to agent α), and w is the range of values describing

[35] In our case, the one-shot interaction case is experienced when $\kappa = 0$ where the agents can only rely on reputation information. Thus, the effect of CREDIT on outcomes of one shot interactions can also be viewed as every point in figures 5.5 or 5.6 for example.

[36] The negotiation power is here defined as the ability of an agent to shift the agreement to a given point in the intersection of the negotiation ranges. The higher its negotiation power, the higher is the utility of the agreement for the agent (and conversely for lower power).

the intersection of the negotiation ranges. Therefore, given a known v^{β}_{min} and v^{α}_{max}, the agreement (i.e. $v^{\beta}_{min} + w'$) can be obtained.

The negotiation power tries to capture different bargaining behaviours that agents may have. Thus, here we do not define what strategies agents may use to influence and reach the agreement but only specify where this agreement will lie (i.e. what utility they will bring) for agents with different (or same) negotiation power resulting from their negotiation strategy (e.g. Boulware and Conceder strategies will concede less and more respectively in a negotiation encounter and hence, have different negotiation powers). In so doing, we focus the analysis on the impact of trust on the agreements agents reach rather than on the bargaining strategies agents might use alongside CREDIT.

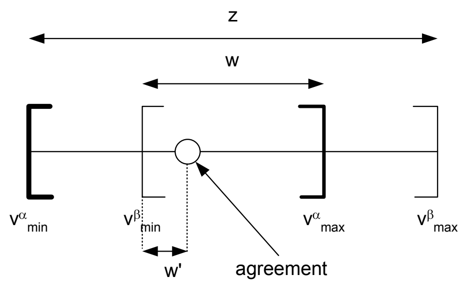


FIGURE 5.4: Deriving the degree of alignment λ and negotiation power ϕ from the negotiation ranges of the agents α and β with ranges $[v^{\alpha}_{min}, v^{\alpha}_{max}]$ and $[v^{\beta}_{min}, v^{\beta}_{max}]$ respectively.

Given the above definitions we start by testing the model with extreme strategies: P v/s P and N v/s P, since these are the basic behaviours that TFT and $STDefect$ implement in different ways to adapt to and exploit their opponent respectively. In so doing, we aim to show that CREDIT can indeed distinguish between trustworthy (i.e. P) and untrustworthy agents (i.e. N) and that it will adapt the agents' negotiation strategies accordingly. In this experiment, the initial trust (based on reputation) of the pairs of agents was varied between 0 and 1. This was achieved by setting different reputation levels on each fuzzy set (i.e. Bad, Average, and Good) for each issue (e.g. for $T = 0.96$, $Rep(s, Bad) = 0.01, Rep(s, Average) = 0.96, Rep(s, Good) = 0)$). θ_{min} was set at 50. The negotiation power was kept at $\phi = 0.5$, while the degree of alignment was set at $\lambda = 0.5$ as well. Thus, by fixing the negotiation power and degree of alignment, we determine a fixed point of agreement between two agents. However, if the negotiation ranges of the agents is modified by CREDIT (see section 5.3.2.1), this point of agreement changes for subsequent negotiation encounters.

With these settings and using equations to derive the trust value and utilities in tables 5.7 and 5.8 and equations 5.7 and 5.11, the agents can reach agreements until the P agent's trust reaches 0.83. At this point, the P agent's negotiation ranges are shifted by the maximum possible for all issues and therefore no agreements are then possible. This is because, the expected utility deviations are such that they extend the (expected) enacted range of values beyond the acceptable range of values (see procedure in equation

5.11). In what follows, we discuss the observations made using these settings.

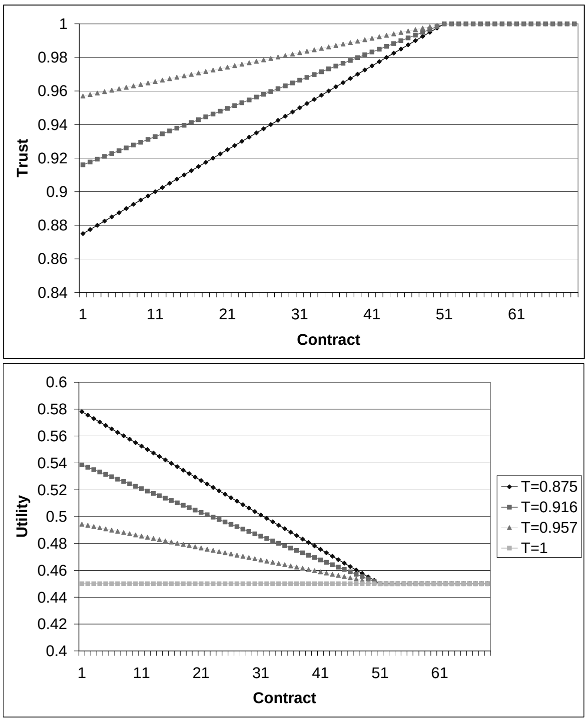


FIGURE 5.5: Trust and Utility of a P-SP faced with a P-SU agent (the baseline at Utility = 0.45 represents the utility of a contract signed when trust = 1).

First, when a P-SU and a P-SP agent are made to negotiate and execute contracts with each other, it was seen that the agents would come to agree on all negotiated contracts. The trust held by the two agents settled at 1.0 throughout the experiment for different starting values of reputation as shown in figure 5.5. As expected, below a trust of 0.83 (i.e. the equivalent confidence level set for all the labels of each issue handled by the SP agent), there is no possible agreement. Moreover, the high trust reached in the long run, enlarges negotiation ranges up to their maximum (with respect to the alignment) and reduces the number of issues to be negotiated down to 4. These factors make way for more agreements. However, with high trust, the agents do not shrink their negotiation ranges such that agreements are made in more relaxed negotiation ranges (given $\phi = 0.5$). This is why the utility of contracts made by the P-SP agent decreases as trust increases. Also, the gradient of the different trust lines are dependent on the difference between confidence levels and reputation levels. Hence, the larger the difference (i.e. the lower the reputation), the larger is the gradient (see equation 5.4).

We then set an N-SU agent against a P-SP agent.[37] The impact of trust on the agreements reached is shown in figure 5.6. As can be seen, if the agent has an initial trust of 1.0 (i.e. all issues have high confidence levels based on reputation), the cut-off trust value of 0.83 is reached as the effect of reputation tails off (given $\theta_{min} = 50$). At

[37]Similar results were observed when the roles of the agents were reversed and it is only as a matter of convenience and succintness that we choose to mention SU v/s SP interactions.

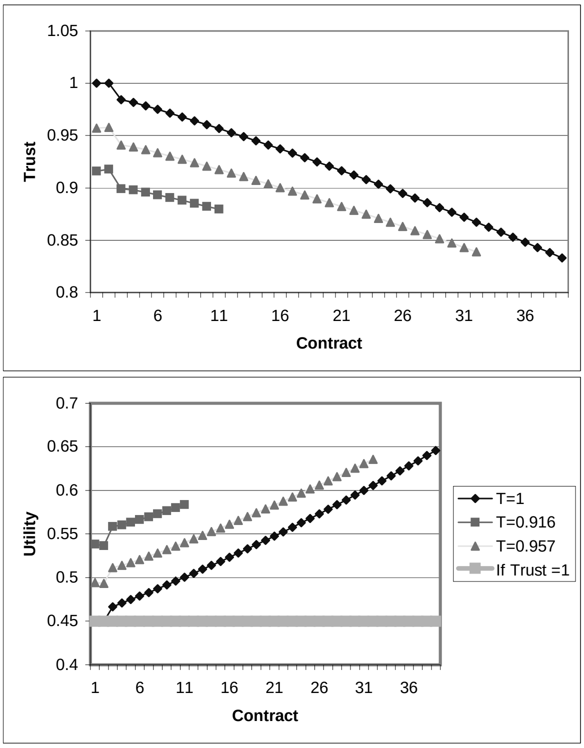


FIGURE 5.6: Trust and Utility of a P-SP faced with a N-SU agent with different starting trust values.

this value of trust, no more interactions are possible between the two agents. Moreover, the value of deals made for the P-SP agent increases as trust decreases. This is because the P-SP agent negotiates for higher prices and shorter time of payment, for example, as trust decreases (i.e. as the confidence levels settle at low values). In so doing, it is decreasing the negotiation range until no further agreements can be reached. This is why all interactions stop after the $39^{th}, 32^{nd}$ and 11^{th} interaction when the initial trust is 1.0, 0.957, and 0.91 respectively. These initial trust values are obtained by setting the reputation on the issues handled by the opponent at different values. Thus for a trust of 1.0, a high level of reputation is set for all issues (i.e. $Rep(x, Average) = 1, Rep(x, Bad) = 0, Rep(x, Good) = 0$ where $x \in \{c, s, qos\}$). For $T = 0.95$, the reputation levels are the same for all issues as well and are set such that $Rep(x, Bad) = 0.05, Rep(x, Average) = 0.99, Rep(x, Good) = 0$ and the trust value is calculated as per equation 5.8 and table 5.8. On the other hand, for $T = 0.91$, the following values are given to each issue in order to simulate the fact that least important issues (in the trust function) have lower reputation levels (i.e. an opponent defects most of the time on the least important issues): $Rep(c, Bad) = 0.07, Rep(c, Average) = 0.99, Rep(c, Good) = 0.1$, used for the most weighted issue c in the trust function, while $Rep(s, Bad) = 0.1, Rep(s, Average) = 0.95, Rep(s, Good = 0)$ for the second most weighted issue s and $Rep(qos, Bad) = 0.15, Rep(qos, Average) = 0.94, Rep(qos, Good) = 0$ for the least weighted issue qos.

The number of agreements reached also depends on the reputation based trust (or initial trust) since it determines what the initial negotiation ranges are. Thus, the lower the initial trust, the smaller the negotiation ranges will be (i.e. the negotiation range is shrunk so as to lead to more profitable contracts as described in section 5.3.2.1). This leads to higher utility agreements (for the P-SP) agent being achieved. A defection by an opponent therefore causes a higher utility loss when the trust in it is initially low than the utility loss we obtain when trust is initially high (i.e. when contracts may be of lower utility). Thus, the P-SP agent shrinks its negotiation ranges faster when the initial trust is low than when it is high and therefore comes to less agreements. Moreover, depending on the weighting of the issue on which an opponent is less trustworthy (see table 5.8), the defections on different issues will cause an equivalently weighted change to the trust value. Thus, even if trust starts at 0.91, those issues which have a low weight in the trust function will already have low confidence levels such that many agreements are not possible and the cut-off trust value only reaches 0.87. Thus, defections on less important issues cause negotiations to end in disagreements even though the *overall* trust value might still be high.

The above results lead us to expect that, in any case, nasty agents will be avoided in the long run by all other types of agents which negotiate with them. The avoidance is made at negotiation time rather than when selecting the agent for an interaction (i.e. before negotiation). If the agent is avoided at selection time (i.e. another agent is selected instead), then it is not given any opportunity to prove its trustworthiness. Selection at negotiation time actually gives a chance to a nasty agent to be trustworthy (i.e. by changing its execution strategy) in order to be accepted in the future (here its execution strategy is fixed to be nasty). As a result, we postulate the following hypothesis:

Hypothesis 2. Untrustworthy agents (i.e. those attracting low trust) achieve fewer agreements than trustworthy ones.

In order to test this hypothesis, we analysed the number of agreements reached by pairs of SU and SP agents using different strategies and for different degrees of alignment between the negotiation ranges of the two parties. Here, we alter the alignment between the negotiation ranges since this alignment is modified whenever agents are deemed untrustworthy and this can, in turn, influence the number of successful agreements achieved. The negotiation power was set at $\phi = 0.5$ and the initial trust based on reputation was set to be 1.0 by fixing the confidence level on each issue to 1 for the set 'Average' and zero for the others (we discuss how altering these values can change the results later). It is to be noted that with the initial trust set at 1.0, agents will be bound to make a number of interactions before their personal measures of trust take over. Using a high initial level of trust also allows us to study the worst case scenario where an agent has a wrong perception of its opponent given the information it gets from society and then uses CREDIT to learn the real behaviour of its opponents.

	Alignment (λ)				
Pairing	**0.1**	**0.25**	**0.5**	**0.75**	**0.9**
NSU-NSP	8	21	39	80	494
NSU-PSP	8	21	39	80	494
NSU-STDefectSP	8	21	39	80	494
NSU-TFTSP	8	21	39	80	494
PSU-NSP	8	21	39	80	494
PSU-PSP	all	all	all	all	all
PSU-TFTSP	all	all	all	all	all
TFTSU-TFTSP	all	all	all	all	all
TFTSU-PSP	all	all	all	all	all
PSU-STDefectSP	16	40	all	all	all
STDefectSU-NSP	8	21	39	80	494
STDefectSU-PSP	16	40	all	all	all
STDefectSU-STDefectSP	16	40	all	all	all
STDefectSU-TFTSP	16	40	all	all	all
TFTSU-NSP	8	21	39	80	494
TFTSU-STDefectSP	16	40	all	all	all

TABLE 5.11: Number of successful negotiations achieved given a varying degree of alignment λ. $\lambda = 0$ means that the negotiation ranges of the two agents have no intersection (i.e. no agreement possible) and $\lambda = 1$ means that they always have an intersection (i.e. all agreements possible). The word '*all*' means that the pair of agents concerned can find an agreement for *any* number of interactions simulated (in practice, however, they may not reach an agreement due to imperfections in the negotiation strategies).

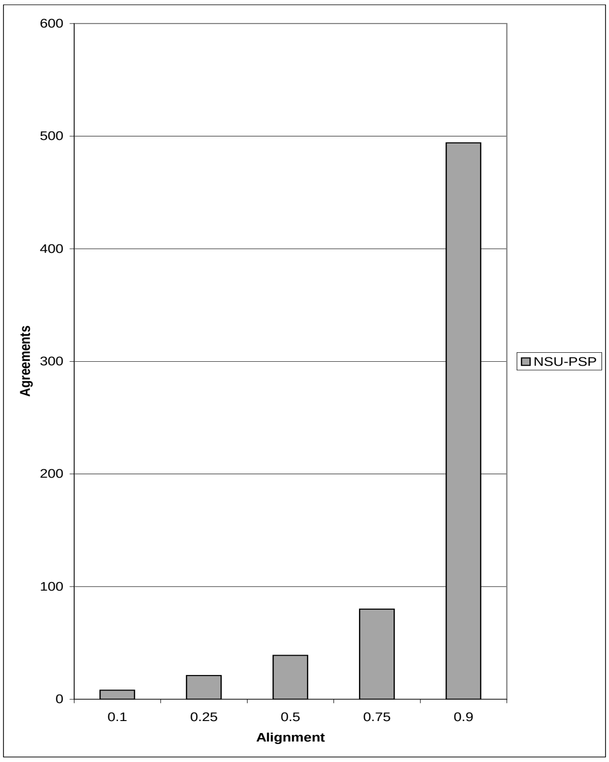


FIGURE 5.7: Number of agreements reached by N faced with P (similar results are obtained with TFT or $STDefect$ as opponents).

The results of the experiments are shown in table 5.11 and the main observations are as follows:

- Whenever the alignment of negotiation ranges $\lambda \leq 0.5$, pairs of P agents and TFT agents still manage to reach agreements. This is because these pairs maintain high trust and always find a coinciding point to agree upon (TFT behaves the same as P if faced with P and TFT). When TFT is faced with N or $STDefect$ in any pairing of SU and SP, the TFT will react in the same way as the N strategy after the first interaction, resulting in the agents achieving fewer agreements.

- Whenever N or $STDefect$ agents are interacting with other or same agents, the number of agreements reached increases with the degree of alignment. This is because, given maximal defections by strategies such as N, $STDefect$, and TFT, the rate of reduction of the negotiation range by CREDIT changes with respect to the size of the alignment. Thus, the higher the alignment, the higher the rate of reduction. This is because the defection sensed is dependent on the agreement point chosen in the intersection which, in turn, depends on the size of the alignment (as determined in equation 5.14 and explained in section 5.4.3.2) . However, this rate of reduction is always less than the rate of increase in the intersection of negotiation ranges (since the agreement point lies *midway* due to $\phi = 0.5$). Therefore, agents will negotiate with defectors more when the alignment is increased.

- When an N agent interacts with P, $STDefect$, or TFT agents, the increase in the number of agreements is quadratic with respect to the alignment degree as shown on figure 5.7. The relationship is quadratic since the calculation of the negotiation ranges is dependent on the calculation of the confidence interval which is itself quadratic with respect to the intersection of negotiation ranges.[38]

- For values of $\lambda > 0.25$, $STDefect$ agents are always interacted with since the trust level in that type of agent remains sufficiently high, at 0.88, (since it defects and cooperates alternatingly) to keep negotiation ranges intersecting when the degree of alignment is moderately large. While this allows the $STDefect$ agent to exploit its opponent when it defects, it also allows other agents to make profitable contracts with it whenever it cooperates (rather than completely avoiding it). Moreover, given that the contracts agreed upon have a higher value for $STDefect$'s opponent (given negotiation ranges are shrunk because of an expected defection), the latter is able to make an additional profit with expected value $(0.58 - 0.45) \times 0.5 = 0.065$ (where 0.58 and 0.45 are the values of the contract when trust is 0.88 and 1

[38] The calculation of the confidence interval involves using the square root of the number of samples under consideration (in approximating to a normal distribution as described in section 5.2.3.2). Increasing the range of samples (or degree of alignment) means that to achieve an equivalent confidence interval, which closes the intersection for a smaller degree of alignment, would require squaring the number of agreements. Hence the relationship is quadratic.

Pairing	**Negotiation Power (ϕ)**				
	1	**0.75**	**0.5**	**0.25**	**0**
NSU-NSP	19	20	21	20	19
NSU-PSP	24	23	21	20	19
NSU-STDefectSP	24	23	21	20	19
NSU-TFTSP	19	20	21	20	19
PSU-NSP	24	23	21	20	19
PSU-PSP	all	all	all	all	all
PSU-TFTSP	all	all	all	all	all
TFTSU-TFTSP	all	all	all	all	all
TFTSU-PSP	all	all	all	all	all
PSU-STDefectSP	36	38	40	42	46
STDefectSU-NSP	19	20	21	23	24
STDefectSU-PSP	46	42	40	38	36
STDefectSU-STDefectSP	36	38	40	38	36
STDefectSU-TFTSP	36	38	40	38	36
TFTSU-NSP	19	20	21	20	19
TFTSU-STDefectSP	36	38	40	38	36

Table 5.12: Number of successful negotiations achieved when varying the negotiation power. $\phi = 0$ means that the SU agent has no power at all and 1 means that the SU agent is always able to achieve the most profitable values for its issues.

respectively and 0.5 captures the fact that $STDefect$ cooperates half of the time) since the $STDefect$ *always* achieves the agreed value whenever it *cooperates*![39]

Given that the utility loss detected is dependent on the value of the initial agreement, we would expect it to change according to the negotiation power the agents have. Given this, and the above observations regarding defectors and the number of agreements reached, we postulate the following hypothesis.

Hypothesis 3. When a high negotiation power agent is faced with an untrustworthy one, the pair of agents will come to fewer agreements than when the negotiation power is low.

To test this hypothesis, we fix the degree of alignment of negotiation ranges to 0.25 (to observe the effect on $STDefect$ and other strategies) and see whether varying the negotiation power alters the number of agreements reached. As can be seen in table 5.12, the negotiation power directly affects the number of agreements achieved particularly

[39]CREDIT is slower to react to defections that occur after a past number of cooperative interactions. Such an event may happen when the opponent knows that it is the last time it is going to interact (i.e. the endgame). The slowness of CREDIT is due to the positive utility loss element being added to an already long list of non-positive utility loss elements of a sample of utility deviations. As a result, the mean of the sample will not be very much affected. Thus, the larger the number of non-positive utility loss elements in the sample the less will be the change in the sample mean when a positive utility loss element is added. Moreover, the behaviour of CREDIT could be made more sensitive to variations over time by tightening the window over the history of interactions is analysed and giving more importance to latest interactions than older ones as in the REGRET system.

when agents are faced with N, $STDefect$ or TFT. Thus, the higher the negotiation power of a trustworthy agent when faced with an untrustworthy one, the lower the number of agreements reached. This is because the higher negotiation power of the trustworthy agent causes the agreement to settle at a higher utility for the trustworthy agent than when its power is lower. Thus, a defection from its opponent is sensed as a stronger defection than when its negotiation power is lower. This leads to a faster reduction of trust and negotiation ranges and fewer agreements.

To understand how both the alignment of negotiation ranges λ and the negotiation power ϕ can affect the number of agreements reached, ϕ and λ were varied in an experiment involving a N-SU v/s P-SP agent[40] and the results plotted in figure 5.8. The reputation of both agents was initially set to be 1. As can be seen, there exists a quadratic relationship between the number of agreements and both the negotiation power and alignment when low values of trust exist in the interaction partners. For very high alignments and low negotiation power, the trustworthy side will take more interactions to see that its opponent is untrustworthy while if the alignment is small and the negotiation power is high, it will take less interactions to do so. This is because the larger the alignment and smaller the negotiation power, the smaller are the defections sensed. This is because contracts made under such conditions will already be of low value to the trustworthy agent (since the opponent has a higher negotiation power). Thus, when the opponent defects, the utility loss sensed is not high enough to significantly affect the trust value. On the other hand, the larger the negotiation power of the trustworthy agent the more utility the contracts will have for the latter. Consequently, defections by its opponent will result in significant utility losses (such that trust is significantly affected). In this case, the smaller the degree of alignment of negotiation ranges, the quicker will be the reduction of the negotiation ranges. Moreover, it is noted that the degree of alignment has a greater importance in determining the number of agreements reached than the negotiation power. This is because the alignment of negotiation ranges determines the space of possible agreements while the negotiation power only changes the point at which the agreement is made in that space (as seen from equations 5.14 and 5.15). Thus, when negotiation ranges are shrunk due to low trust, the negotiation power barely changes the value of agreement (particularly when ϕ is high) while the alignment, which itself determines by how much negotiation ranges can be shrunk, significantly affects the point at which the agreement is made.

5.4.4 Experimental Set 2: Facing Degree Defectors

Having investigated CREDIT's behaviour with different opponents with extreme strategies in the previous section, we now aim to test how CREDIT can manage with oppo-

[40] The choice of these strategies for the experiment is only made to simplify the analysis. We expect the same properties to be exhibited with other pairings since the shape of the curves are independent of the strategies, although the actual intercepts may not be the same.

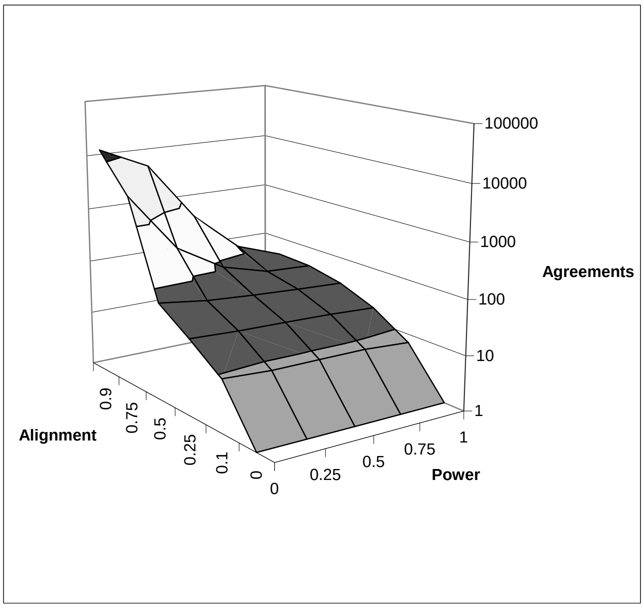


FIGURE 5.8: Number of agreements reached between a N v/s P for various alignment degrees and negotiation power.

nents which do not always defect maximally, but rather defect by a given degree. This is important because agents may not always be faced with opponents that always cooperate or defect. Indeed, the performance of the opponent might lie in-between these two extreme strategies. For example, an SP agent may defect to some degreee by unconditionally charging a transaction fee for supplying bandwidth even though this may not be included in the contract. Thus the SU agents could find themselves paying a price that lies outside their set of acceptable prices. An SU agent might also be paying for the bandwidth by posting a cheque such that it always adds two more days to t_c. The SP agent may therefore find itself getting the money after its acceptable deadline. In general, we believe a trust model should be able to adjust the behaviour of an agent such that it is still able to come to some form of agreement with such defectors. Thus, it should allow the agent to *still make profitable contracts* (e.g. by specifying more stringent conditions or penalties that cover the losses) when evolving in environments where not all agents perform contracts perfectly (or are perceived as doing so).[41]

We focus on experiments where only one of the two agents defect.[42] This allows us to focus on the influence of CREDIT on the negotiation and rule out other possibilities for the observations made. First we test to see if CREDIT is able to recognise such defectors with varying degrees of alignment of the interacting agents' negotiation ranges. The agents are first given high reputation (i.e. trust value 1.0) and the agents have

[41]Generally current trust models, such as (Sen et al., 2000; Birk, 2001) or (Yu and Singh, 2002b), do not cater for this type of defection. Those models which do (e.g. REGRET (Sabater and Sierra, 2002), Schillo et al. (Schillo et al., 2000)) do not use the information about degree defectors in any significant way apart from partner selection.

[42]We report experiments where the SU agents are defectors. However, the behaviour of CREDIT does not change if we reverse the roles.

Degree of Defection	**Alignment(λ)**				
	0.1	**0.25**	**0.5**	**0.75**	**0.9**
0.2	all	all	all	all	all
0.4	11	30	all	all	all
0.6	8	21	42	all	all
0.8	8	21	39	118	all
0.9	8	21	39	80	all

TABLE 5.13: Number of agreements reached with degree defectors with different degrees of alignment of negotiation ranges.

equal negotiation power. Thus, initially agents are expected to negotiate with relaxed negotiation ranges until they assimilate the trustworthiness of their opponent. In so doing, we see whether the agents will avoid a degree defector if the degree of defection is too high for the negotiation ranges afforded by the trustworthy agent. To this end, table 5.13 records the number of agreements reached for different degrees of alignment. As can be seen, the number of agreements reached by the defectors increases with increasing degrees of alignment. This is because the agents have larger negotiation ranges and therefore they are able to adjust them sufficiently such that the higher degrees of defection still end up being profitable. Given this, we postulate the following hypothesis:

Hypothesis 4. For different degrees of alignment of negotiation ranges, an agent using CREDIT is able to adjust its negotiation ranges to prevent defections by its opponent lying outside its acceptable regions.

To check if CREDIT is always able to profitably adjust negotiation ranges, we recorded the difference between the utility of the executed contract and the value of the contract for minimum acceptable values of the issues the opponent handles in the contract. If the result is negative, the enacted values lie outside the acceptable ranges and if it is positive or zero, then the enacted values lie within the acceptable ranges. We study a pair consisting of a P-SP agent and a $DD_{0.2}$-SU agent with different degrees of alignment.[43] The graphs obtained for other degrees of defection were found to be similar in nature for various degrees of alignment.

As can be seen, when the alignment is low (0.1 and 0.25), the P agent suffers some utility loss for some part of the deals (i.e. 23 and 35 deals respectively). During these interactions, the high reputation contributes to the high trust in the defector (since

[43] $DD_{0.2}$ means that the agent defects by a degree of defection of 0.2 on all issues it handles. This means that the agent always pays at most 4 days late (i.e $0.2 \times (30 - 10)$), abides by a security level which is at most 1 level lower (i.e $0.2 \times (6 - 1)$), and uses the connection at a rate of 19 connections/min at most (i.e. $0.2 \times (100 - 5)$) more than agreed (where the multipliers represent the range of values for the issues concerned). We choose a degree of defection of 0.2 to provide a better analysis of the effect of the degree of alignment on the behaviour of CREDIT when faced with degree defectors (since CREDIT achieves all possible agreements with $DD_0.2$ as seen in table 5.13).

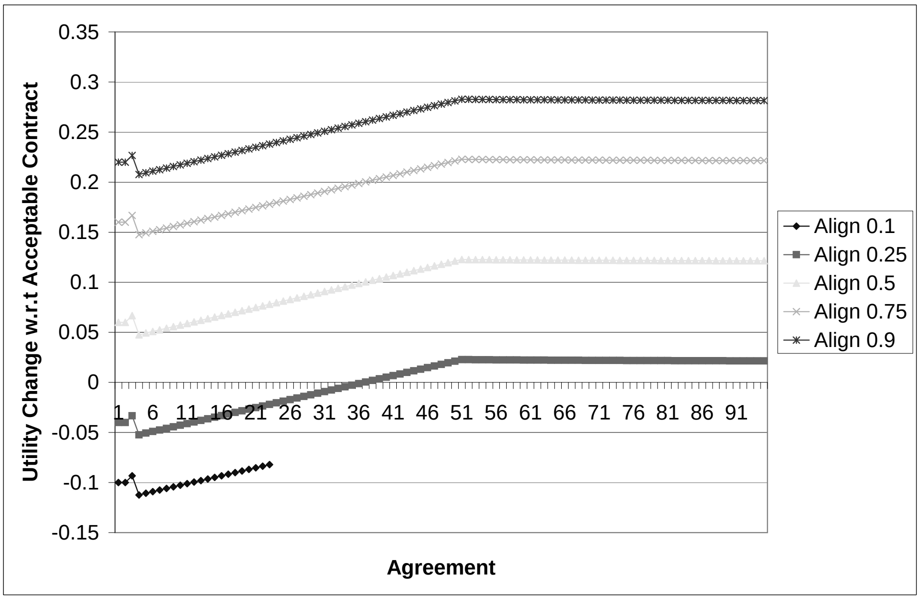


FIGURE 5.9: Utility change for P-SP agent (v/s $DD_{0.2}$-SU) after execution of contracts relative to acceptable contracts for different levels of alignment.

$|CB_{\alpha,\beta}| < \theta_{min} = 50$) and this causes negotiation ranges to be relaxed for some interactions. However, after having adjusted its negotiation ranges according to its own confidence levels in the defector (i.e. $|CB_{\alpha,\beta}| > \theta_{min}$), the P agent suffers no more utility losses. Moreover, the utility of signed contracts is seen to rise with a larger alignment of the negotiation ranges, such that the P agent is able to increase its utility gain. This leads us to postulate the following hypothesis:

Hypothesis 5. When facing increasing degrees of defection, an agent using CREDIT is able to adjust the negotiation ranges so as to engage in profitable interactions.

Having tested how CREDIT fares with different degrees of alignment in the previous set of experiments, we now wish to see how it is able to cope with increasing degrees of defection for a given fixed degree of alignment. From table 5.13 we can see that for degrees of alignment below 0.75, the number of agreements reached by high degree defectors (i.e. 0.8 or 0.9) is less than for low degree defectors. While this provides evidence that high degree defectors will be avoided in the long run, for higher degrees of alignment it is not apparent that CREDIT still manages to enforce profitable interactions. To test for this property, we recorded the utility gain with respect to acceptable contracts (as in the previous experiment) for different degrees of defection (see figure 5.10). Here, the agents are all assumed to have high reputation and the degree of alignment is set to 0.75. As can be seen, CREDIT is indeed able to profitably adjust negotiation ranges against low degree defectors or else it shrinks the negotiation ranges sufficiently so as to prevent agreements from being reached with high degree defectors. This validates our hypothesis.[44] Moreover, note that given our definition of degree of defection (see section

[44]It is also to be noted that all the models suffer a jump in utility losses on the first few interactions.

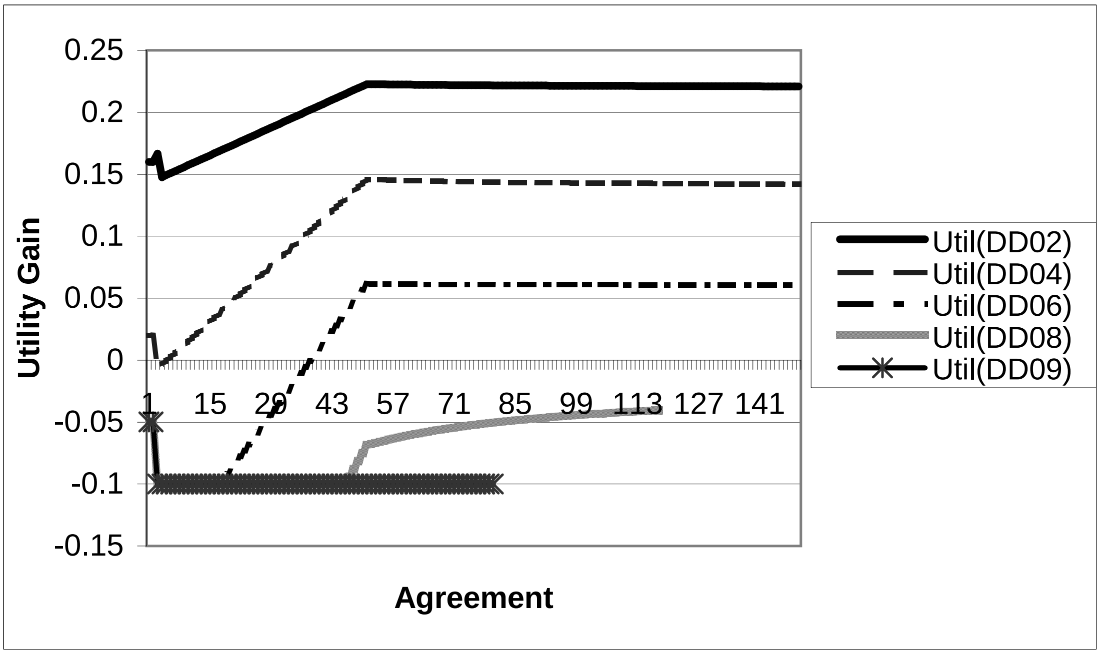


FIGURE 5.10: Utility change for P-SP agent (v/s DD_n-SU), for $n = \{0.2, 0.4, 0.6, 0.8, 0.9\}$, after execution of contracts relative to acceptable contracts for different levels of alignment.

5.4.2), if values of a contract are close to the edges of the negotiation range, the degree defectors cannot be easily differentiated from each other.

It should also be noted that the utility gain for the 0.9 defector stays at -0.1 for all deals made. This is because this agent defects so as to achieve a contract that is at most equal to 90% of the maximum defection possible on a given issue and is therefore avoided after a small number of interactions (80 as in the case of nasty agents in table 5.11). Thus for values contracted that lie closer to the maximum degree of defection, the 0.9 defector will act in the same way as a nasty agent. For example, a 0.9 defector SU agent will defect to the latest possible time of payment (i.e. 30 days) if t_c has been contracted for 29 days. Otherwise, if t_c has been agreed for 30 days, a 0.9 defector SU agent will only defect to 30 days (i.e. 90% of the interval). Thus for a high degree of alignment such as $\lambda = 0.9$ (when the 0.9 defector achieves all contracts), we expect CREDIT to force the enacted contract to lie as close as possible to the range of acceptable values in the long run. This should make the utility losses with respect to acceptable contracts tend to zero. This is confirmed by the graph in figure 5.11.

This is due to the rule of the P-SU agent firing (see table 5.4) such that the qos is added to the set of negotiation issues. Thus a defection on the qos causes the utility losses experience to start at an even lower value.

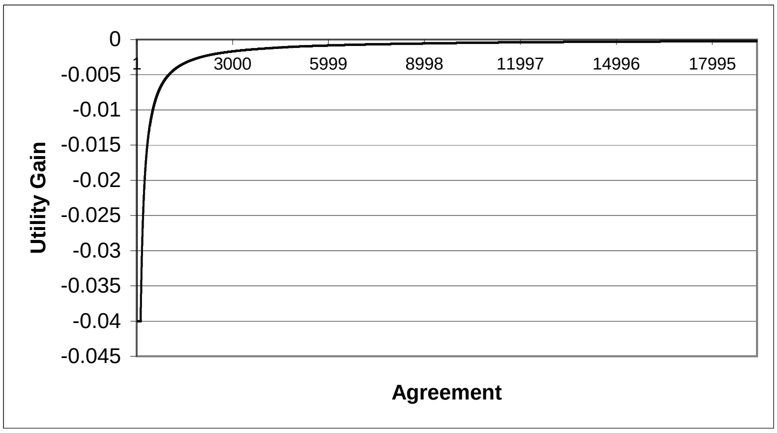


FIGURE 5.11: Utility change for P-SP agent (v/s $DD_{0.9}$-SU), after execution of contracts relative to acceptable contracts for $\lambda = 0.9$.

5.5 Benchmarking CREDIT

Having analysed its performance in isolation, the next step is to put CREDIT's into context. To this end, we compare CREDIT's effectiveness against other comparable models that are available in the literature. The most relevant models are those by Witkowski et al. and Marsh (see section 3.1 for more details). We will refer to Witkowski's model as WT and Marsh's model as MT from now on. In general, these two models calculate trust by analysing an opponent's behaviour during interactions in a similar manner to CREDIT (others base trust on reputation and assume an analysis of interactions (Sabater and Sierra, 2002; Yu and Singh, 2002a; Mui et al., 2002)). Even though these models do not specifically use trust to influence negotiations (which we will apply them to in our benchmarks), we use them as representative of two classes of models each having their own merits (as we will see later in this section): (i) those models that change trust values by arbitrarily set values; (ii) those that are based on a different model of risk. The formulae used to calculate trust in WT and MT are as follows:

WT:

$$T_{wt}(\alpha,\beta,x) = \left\{ \begin{array}{ll} T_{wt}(\alpha,\beta,x) - \rho \times T_{wt}(\alpha,\beta,x) & defect \\ T_{wt}(\alpha,\beta,x) + \varphi \times (1 - T_{wt}(\alpha,\beta,x)) & cooperate \\ T_{wt}(\alpha,\beta,x) + n \times (1 - T_{wt}(\alpha,\beta,x)) & faced\ with\ DD_n \end{array} \right\} \quad (5.16)$$

where $T_{wt}(\alpha,\beta,x)$ is the trust of α in β over issue x, ρ weighs the impact of a defection on the trust value, φ weighs the impact of a cooperation on the trust value and $n \in [0,1]$ is a defection degree perceived (i.e. when the defector does not completely cooperate nor completely defect). WT updates the trust value after each interaction depending on the behaviour of the opponent. A defection is equivalent to the opponent achieving a less profitable value for an issue such that it causes a utility loss to the agent. Cooperation is equivalent to enacting the exact values of the issues specified in a contract. As can be seen from the above formulae, a given degree of defection is considered as cooperation rather than defection. Therefore an agent defecting by a given degree will *increase* its opponent's trust in it. While we recognise that this equation fits the purpose of the experiments by Witkowski et al., we consider it counterintuitive to the notion of trust (and irrational). Therefore, we adopt only the first two equations in order to capture cooperations and defections.

MT:

$$T_{mt}(\alpha, \beta, x) = U_x^{\alpha}(v) \times \widehat{T}_{mt}(\alpha, \beta, x) \tag{5.17}$$

where $U_x^{\alpha}(v)$ is the utility of issue x with value v for α (including its weight in α's utility of a set of issues including x), $T_{mt}(\alpha, \beta, x)$ is the trust of α in β over issue x. MT updates the trust in a similar manner to our model by considering the risk involved in the interaction given the subjective perception on the opponent. $\widehat{T}_{mt}$ is actually an estimation of trust in the opponent given the risk incurred by an agent. While Marsh in (Marsh, 1994) does not give a concrete implementation for this value in his model, we calculated it through our probabilistic modelling of the opponent as described in section 5.2.3.2. This method respects the reasoning behind the meaning of $\widehat{T}_{mt}$.

WT and MT do not specifically associate trust with negotiation ranges. Therefore, we map the trust values obtained from the above equations to the probabilistic model of utility loss which CREDIT uses to shrink the negotiation ranges. This is achieved by inverting equation 5.8 and applying the multiplying factors (φ and ρ for WT and $U^{\alpha}(x)$ for MT). For WT we also set $\rho = \varphi = 0.25$ (as used in Witkowki's experiments). Moreover, T_{wt} and T_{mt} were used in the norms to specify issues that were to be negotiated or not as specified in section 5.3.2.2.

With the above settings implemented, the models were tested in a similar manner to our model as in section 5.4.2. However, here we focus on those aspects where the trust can directly influence the interaction between two agents.[45] Therefore, we choose the following measures to see how well the models are able to elicit and use trust in order to:

1. Prevent exploitation by a (extreme) defector (e.g. N or $STDefect$)

2. Allow agents to negotiate contracts with agents which defect by degrees (e.g. $DD_{0.2}$ or $DD_{0.4}$).

[45] We do not test how agents will choose their interaction partners using trust derived from different models since this behaviour can be inferred from the results of other experiments which test the model at interaction time.

Each of these is now detailed in turn.

5.5.1 Experimental Set 1: Facing Extreme Strategies

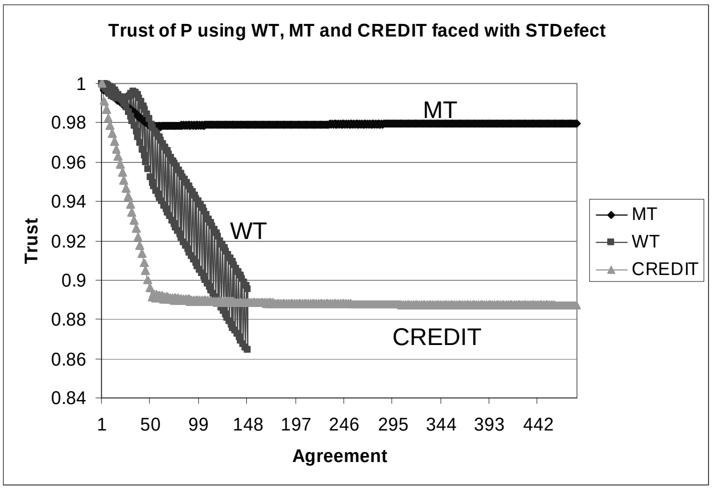
Here, we base our experiments on the best execution strategy to be used against a defecting agent irrespective of the trust model. In so doing, we avoid relating the strategy to the variables to be measured (e.g. a philanthropic agent might do better with CREDIT but a nasty agent could do better with another trust model). Therefore, based on Axelrod's experiments, we choose to use the TFT strategy as the execution strategy of one agent which interacts against a nasty agent or a strategic defector. TFT was then shown to be better than any other strategy at preventing exploitation (but never obtained the highest reward itself).

Given this, our first set of experiments aims to show how well an agent can adapt to a defecting agent. This aims to show how the agent can detect bad behaviour using its trust model and alter its own behaviour at negotiation time. To this end, we plot the trust value of the TFT agent against the number of negotiations the agents go through. The initial level of trust was set at 1 as in the previous experiments by setting the reputation values of the fuzzy set 'Average' to 1 for all issues and 0 for the other sets. The results are shown in figure 5.12.

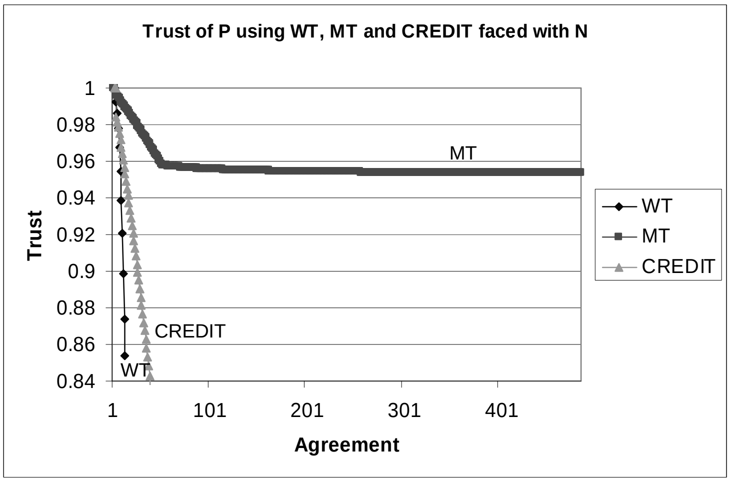
As can be seen, the fastest to react to defections by a nasty agent is WT. Trust goes down within the first few interactions settling at 0.83 for a nasty agent and oscillating until it reaches 0.867 for a strategic defector. CREDIT gradually settles at a value of 0.83 for a nasty agent (after which the agents do not interact) and a value of 0.88 for a strategic defector. This occurs after around 40 interactions for the nasty agent and 60 interactions for the $STDefect$ agent. MT, which reacts slower than CREDIT, decreases the trust to a value of 0.952 for the nasty agent and 0.98 for the $STDefect$ agent after the first 50 interactions over which the defectors' high reputation prevails.

These results are explained by the formulae used to calculate trust values. According to equation 5.16, WT decreases or increases trust from 1 by a factor of 0.25. This is an ad-hoc method of manipulating the trust value. This simple heuristic is based on applying punitive action for bad behaviour and rewarding for good behaviour (good meaning cooperation). It effectively reduces trust by a relatively large amount (asymptotically reaching 0 or 1) compared to the other models. Moreover, the values of trust in strategic defectors and nasty agents are lower than those reached in CREDIT and MT. The latter models also react more slowly (i.e. they need more interactions to get a reliable rating) to bad behaviour. This is because they are both based upon a statistical analysis of the behaviour of an opponent which takes a larger number of interactions to be precise, while it takes only 20 interactions for WT (60 for CREDIT and MT).

It is to be noted that MT never stops negotiating with both the nasty agent and the



(a) Trust models faced with $STDefect$.



(b) Trust models faced with N.

FIGURE 5.12: Plots showing trust values of a TFT-SU agent v/s $STDefect$-SP and N-SP agents for CREDIT, WT, and MT respectively. For MT, each negotiation it undertakes reaches an agreement meaning that it never stops negotiating with a defector.

strategic defector and always settles on an agreement with them. Therefore, we can infer that multiplying the utility (or weight) with the trust value (as in equation 5.17) reduces the effect of defections on the trust value, allowing better exploitation by an opponent (i.e. the nasty agent). Moreover, the weights reduce the influence of trust on negotiations since the loss in confidence is reduced whenever a defection occurs.

Instead, CREDIT gives equal importance to all issues at negotiation time. Thus, CREDIT is able to avoid a nasty agent while still engaging in interactions with the strategic defector (for a degree of alignment of 0.5). In so doing, CREDIT is still able to make profitable contracts in the long run while WT is not able to do so with the $STDefect$ agent (at least for the interactions where the latter cooperates). Given that the strategic defector defects half of the time, the expected utility in any one interaction with it is half the utility of the issue-value pairs agreed upon. Independent of trust models, this expected utility when facing an $STDefect$ agent is $0.5 \times 0.45 = 0.225$ (where 0.45 is the value of a contract when trust is 1). However, CREDIT shrinks negotiation issues since it expects a given 'degree' of defection from the $STDefect$ agent. Thus, when $STDefect$ cooperates, it actually enacts a contract which has a higher value than the expected degree of defection. Therefore, CREDIT induces an expected utility of $0.5 \times 0.58 = 0.29$ (where 0.58 is the utility of the contract for the P agent when negotiation ranges are shrunk when trust is 0.88) and is able to gain more utility than both WT and MT in the long run (for MT, the expected values is $0.5 \times 0.48 = 0.24$).

5.5.2 Experimental Set 2: Facing Degree Defectors

In this set of experiments we aim to compare the performance of the trust models when the latter are used by agents facing opponents which defect by degrees. Here a P agent was made to interact with the defectors with negotiation power of 0.5 and degree of alignment between negotiation ranges of 0.5. The agents using WT and MT were faced with agents defecting with degrees of 0.2, 0.4, 0.6, 0.8, and 0.9. Given the properties we have identified in the previous section, we can expect that WT will reduce trust drastically no matter what the degree of defection of its opponent is. As for MT, we expect it to react to degree defectors more slowly than CREDIT given that it will weigh defections by the utility of the issues it has contracted. To this end table 5.14 shows the results of the experiments carried out.

Degree of Defection	**Trust Model**		
	MT	**WT**	**CREDIT**
0.2	all	13	all
0.4	all	13	all
0.6	all	13	42
0.8	all	13	39
0.9	all	13	39

TABLE 5.14: Number of agreements reached by pairs of P and DD_n agents.

As can be seen, MT actually continues negotiations with all defectors while WT considers all degree defectors to be the same as nasty agents. CREDIT instead considers each degree of defection differently. In order to see how well the agents are able to adapt their negotiation ranges so as to minimise utility losses, we plot the utility gain of the P agent with respect to acceptable contracts (i.e. those for which the minimum acceptable values are achieved by the opponent). Here we show the utility gains of the P agents when faced with a $DD_{0.4}$.

As can be seen, WT shrinks its negotiation ranges too much to allow possible negotiations with degree defectors (for $\lambda = 0.5$ and $\rho = 0.5$). This is because WT detects a given degree of defection as a full defection even for very low degrees of defection. This is explained by its ad-hoc method of calculating the trust value. Indeed, the trust value

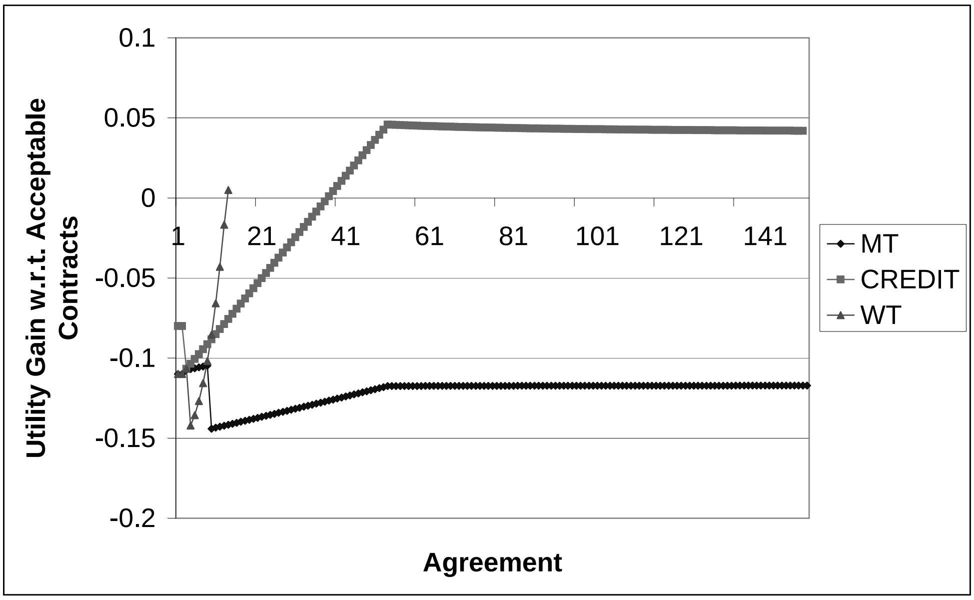


FIGURE 5.13: Plots showing utility gain w.r.t an acceptable contract of P agents v/s $DD_{0.4}$ for MT, CREDIT, and WT.

is not calibrated according to the value achieved by the opponent, but only based on the action of defecting. MT does not perform any better in this sense even for very low degrees of defections since it is unable to sufficiently adjust the negotiation ranges so as to have executed contracts fall within the acceptable range. This is due to the model giving low importance to defections by considering the weights of issues in considering its trust (see equation 5.17. CREDIT shows the best performance overall by achieving profitable deals even with some degree of defection from its opponent (see section 5.4.4 for more details).

5.6 Summary

In this chapter we have detailed a novel trust model called CREDIT and shown it to be both efficient and effective at guiding agents in their interactions given uncertainty about the honesty and reliability of their opponents. Thus, CREDIT meets the initial objectives described in section 1.5. Specifically, we have shown how trust can be related to the expected utility loss in interactions. Here, the combination of confidence levels and reputation using fuzzy sets provides a robust way of combining societal measures with individual measures of trust. Fuzzy sets also take into account the ontological dimension of trust measures that agents may share with each other. Using the trust measures thus devised, we have described an algorithm to calculate trust that is linear with respect to the number of past interactions and incrementally quadratic with respect to the number of fuzzy sets used. Finally, having tested and benchmarked CREDIT we here summarise the conclusions inferred from the observations made.

1. By combining norms and trust, CREDIT is able to reduce the length of the negotiation dialogue (analogous to time) required to reach an agreement (see section 5.4.3.1).

2. CREDIT is able to prevent nasty agents from exploiting philanthropic and tit-for-tat agents by adjusting the agent's negotiation stance. Thus, when nasty agents are encountered, the agent shrinks its negotiation ranges such that no agreement is possible (section 5.4.3.2).

3. CREDIT is better able to engage in profitable contracts with strategic defectors than other trust models (see section 5.5.1). This shows that the model is able to adapt to agents with varying reliability.

4. CREDIT is able to cope with degree defectors better than other trust models. It does so by eliciting profitable contracts in the long run or by avoiding high degree defectors (section 5.4.4).

Thus we have shown how the outcomes reached in bargaining can be significantly improved when CREDIT is used to model the reliability and honesty of interacting agents. In this respect, CREDIT clearly differentiates itself from current work in the area of trust which has, up to now, only considered using trust to select the most reliable or honest partner. Moreover, CREDIT is the only model to provide a comprehensive analysis of the context in which agents interact by considering norms and by using such norms during the bargaining process. Finally, in benchmarking CREDIT, we have provided the first set of guidelines against which future trust models can be benchmarked (in terms of the experiments carried out to differentiate CREDIT from WT and MT).

Through CREDIT we have shown how bargaining agents can use trust to reduce uncertainty about the reliability and honesty of agents. CREDIT also reduces, to some extent, uncertainty about the actions of the agents by shrinking negotiation ranges when defectors are detected. However, CREDIT neglects the uncertainty about the preferences of the agents and usually expands the action set when trustworthy agents are interacted with. To remedy this, we develop the PN model (in chapter 7) that enables agent to further reduce uncertainty about the action set and preferences through the use of arguments. Given this, with the combined use of PN and CREDIT, agents can substantially reduce uncertainty in most aspects of their bargaining encounters. Given this, the (combined) use of these models in a practical application is described in chapter 8.

Furthermore, in the next chapter, we show how a trust model, such as CREDIT, can be integrated into negotiations based on MD techniques. In so doing, we aim to show how trust at the individual level (i.e. CREDIT) can be combined with trust at the system level (as per the requirements described in section 3.3) to elicit efficient outcomes (which bargaining techniques cannot guarantee) while ensuring that the most reliable agents are chosen as per the objective set in section 1.5.

Chapter 6

Trust-Based Mechanism Design

Having demonstrated the significant improvement that CREDIT brings to the utilities that agents gain in bargaining encounters, this chapter focuses on using trust in mechanisms or protocols in order to obtain an efficient partitioning of the resources that agents negotiate. As we have seen in chapter 3, there are a number of system-level trust mechanisms that already enforce some level of honesty (about their costs or valuations) among the participating agents so as to reach efficiency. However, none of these mechanisms effectively select the most reliable agents in determining the partition of resources. To this end, in this chapter we propose the area of TBMD as an extension of traditional MD, where trust is used to select the most reliable agents in determining the outcome of the mechanism as per our objectives. In this way, TBMD effectively reduces the uncertainty surrounding both the reliability and honesty of agents (as per objectives set in section 1.5). In particular, we develop the TBM, as an extension of the Vickrey-Clarkes-Groves (VCG) class of mechanisms, that is incentive compatible, individually rational, and efficient. Thus, in our TBM agents are incentivised to honestly reveal their valuations (or cost) as well as their trustworthiness and the reputation of other agents. Moreover, we show how our TBM is more robust than other comparable mechanisms at dealing with biased ratings from some of the agents (see section 3.3).

The rest of this chapter is structured as follows. In section 6.1, we justify the need to extend traditional MD to consider trust. Section 6.2 describes related work in the area of mechanism design. Section 6.3 shows how a standard VCG mechanism usually determines the outcomes in a task allocation scenario. This serves as the basis for describing TBMD in section 6.4. Section 6.5 describes our TBM and demonstrates how the TBM generalises the standard VCG mechanism to consider trust. Section 6.6 empirically evaluates our TBM and shows how it is indeed effective and efficient at choosing the most reliable agents over repeated encounters (as trust models build clearer trust measures), and that it is more robust than comparable mechanisms against biased ratings. Finally, section 6.7 summarizes the main achievements of this chapter and discusses their relationship to other models we provide in this thesis.

6.1 Introduction

As discussed in section 1.2, MD is the field of microeconomics that studies how to devise systems such that the interactions between strategic, autonomous and rational agents lead to outcomes that have socially-desirable global properties. Given that the designer of MAS typically has many of the same aims, there is a growing body of work that seeks to exploit the tools and concepts of MD to this end (Dash et al., 2003). However, an important facet of MAS that is rarely considered in MD is that agents do not always complete their tasks as planned or promised (this means they are not always *sucessful*). Thus, for example, an agent may not always complete every task it starts or it may default on payment for a good. Furthermore, in traditional MD an agent chooses to interact with partners based on their costs or valuations only. However, cheapest is not always best and these agents may ultimately not be the most successful. Thus, in many practical situations the choice of interaction partners is motivated by an agent's individual model of its counterparts, as well as by information gathered from its environment about them. For example, on eBay buyers determine the credibility of particular sellers by considering their own interaction experiences with them (if they have any) and by referring to the historic evaluated information provided by other buyers. To capture this phenomenon, we exploit the notion of *trust* used in the CREDIT (where trust results from the combination of confidence and reputation) to represent an agent's perception of other agents' reliability. In this chapter, we refer to an agent's reliability as its probability of success (POS) in completing its task. This, in turn, leads us to propose the area of TBMD as an extension of traditional MD that adds trust as an additional factor to costs and valuations in decision making.

As we argue in earlier in the thesis (see section 1.3), the trust in an agent is generally defined as the expectation that it will fulfill what it agrees to do, given its observable actions and information gathered from other agents about it (see section 1.3). By their very nature, different agents are likely to hold different opinions about the trust of a particular agent depending on their experiences and the specifics of the trust model they use (see section 3.1). As a result, we cannot simply extend traditional MD (e.g. the VCG mechanism) to encompass the notion of trust because such work is predicated on the fact that agents have *private and independent* information which determines their choice over outcomes. Trust, on the other hand, implies *public and interdependent information* (see sections 3.1.2 and 3.2.2).

In this work, we specifically consider MD in the context of task allocation (where it has often been applied Sandholm (2003)). In our scenario, agents may have different probabilities of success in completing a task assigned to them (e.g. it may be believed that a particular builder has a 95% chance of making a roof in five days, while another builder may be believed to have a 75% chance of doing so). Moreover, an agent may assign different weights to the reports of other agents depending on the similarity of their types. For example, consider a "repair engine" task assigned to a garage. In this case, two agents owning a Ferrari would assign higher weights to each other's report about the POS of the garage than they would to the report of another agent which owns a Robin Reliant.

Against this background, this chapter develops and evaluates the notion of trust-based mechanism design. We also define the general properties that trust models must exhibit to allow a trust-based mechanism to generate an optimal allocation of tasks. In particular, we advance the state of the art in the following ways:

1. We specify the properties that trust models must satisfy to be incorporated in mechanisms that permit efficient allocations.
2. We generalise the standard VCG mechanism to incorporate the notion of trust.
3. We prove that the trust-based mechanism we develop is efficient, individually rational, and incentive compatible.[1]
4. We empirically show that our trust-based mechanism leads to the most successful and cheapest agents being selected to fulfill an allocation in the long run and that it performs better than comparable mechanisms when agents' reports of POS are biased.

6.2 Related Work

In associating trust to mechanism design, we build upon work in both areas. In the area of trust and reputation, a number of computational models have been developed (see chapter 3 for a review). While these models can help in choosing the most successful agents, they are not shown to generate efficient outcomes in any given mechanism. An exception to this is the work on reputation mechanisms (see section 3.2.2). However, as it was shown in section 3.2.2, these mechanisms only produce efficient outcomes in very constrained scenarios and under strict assumptions (e.g. in (Dellarocas, 2002) sellers are monopolists and each buyer interacts at most once with a seller and in (Jurca and Faltings, 2003a) the majority of agents must already be truthful for the mechanism to work).

In the case of MD, there has been comparatively little work on achieving efficient, incentive-compatible and individually-rational mechanisms that take into account *uncertainty* in general. An exception to this rule is the dAGVA mechanism (Mas-Colell et al.,

[1]The mechanism we develop also forms the only class of mechanisms that have these properties under a Nash equilibrium strategy when factoring trust into the decision making process. Intuitively, this follows from the uniqueness of the VCG which charges agents their marginal contribution to the system. Since we use a similar technique to develop our mechanism we believe the same result will ensue (the formal proof of this assertion is beyond the scope of this thesis).

1995) which considers the case when the types of agents are unknown to themselves but are drawn from a probability distribution of types which is common knowledge to all agents. However, in our case, the agents know their types and these incorporates their uncertainty related to fulfilling a task. Porter et al. (Porter et al., 2002) have also considered this case and their mechanism is the one that is most closely related to ours. However, they limit themselves to the case where agents can only report on their own POS. This is a drawback because it assumes the agents can measure their own POS accurately and it does not consider the case where this measure may be biased (i.e. different agents perceive the success of the same event differently). Thus our mechanism is a generalisation of theirs (see section 6.5.2 for the formal proof).

Finally, our work may also seem to be a case of interdependent, multidimensional allocation schemes (Dasgupta and Maskin, 2000) where there is an important impossibility result of not being able to achieve efficiency when considering interdependent, multidimensional signals (Jehiel and Moldovanu, 2001). However, we circumvent this by relating the trust values to a probability that an allocation is completed, rather than to an absolute valuation or cost signal.

6.3 A Standard VCG Task Allocation Scheme

In the rest of this chapter, we use a different notation from that presented in chapter 4 so as to conform to the usual notation used in the domain of mechanism design. Given this, we consider a set of agents $\mathcal{I}$, where $\mathcal{I} = \{1, \ldots, I\}$, and a set of possible tasks $\mathcal{T}$. Each agent $i \in \mathcal{I}$ has a particular value, $v_i(\tau, \theta_i)$, for having a task (completed by another agent), $\tau \in \mathcal{T}$, which is dependent on its type θ_i drawn from a possible set of types, Θ_i. An agent i also has a cost, $c_i(\tau, \theta_i)$, of attempting to complete a task. Given a vector of values, $\boldsymbol{v}(\tau, \theta)$, and costs, $\boldsymbol{c}(\tau, \theta)$, from the set of agents, we can determine the value of an allocation $K \in \mathcal{K}$ where $\mathcal{K}$ is the set of all possible mappings of $\mathcal{T}$ to $\mathcal{I}$. Once a certain allocation K is implemented, an agent i is then asked to pay for the task(s) it requested or receive payment for the task(s) it performed. The overall transfer of money to a particular agent i is denoted by r_i. As is common in this domain, we assume that an agent is rational (expected utility maximiser) and has a quasi-linear utility function (Mas-Colell et al., 1995). The following definition of the utility function refines our earlier definition in chapter 4 to take into account costs as well as valuations of tasks/issues:

Definition 1. A **quasi-linear utility function** is one that can be expressed as:

$$u_i(K, r_i, \theta_i) = v_i(K, \theta_i) - c_i(K, \theta_i) + r_i \tag{6.1}$$

In devising a mechanism for task allocation, we focus on *incentive compatible direct revelation* mechanisms (DRMs) by invoking the *revelation principle* which states that

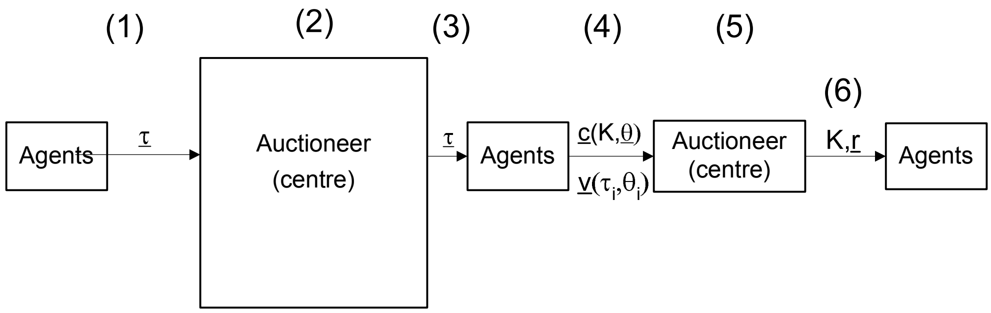


FIGURE 6.1: Simple task allocation model using the VCG mechanism.

any mechanism can be transformed into a DRM. In this context, "direct revelation" means the strategy space (i.e. all possible actions) of the agents is restricted to reporting their types and "incentive compatible" means the equilibrium strategy (i.e best strategy under a certain equilibrium concept) is truth-telling. Hence, in our allocation scheme, the agents report their types to a centre which then decides on the allocation K and the reward vector $\boldsymbol{r}$ and reports these back to the agents. The problem at hand is then to find a mechanism $M(\boldsymbol{v}(\tau, \boldsymbol{\theta}), \boldsymbol{c}(\tau, \boldsymbol{\theta})) = \{K, \boldsymbol{r}\}$ that fulfills the following commonly sought objectives in MD:

- *Efficiency*: an allocation that maximises the total utility of all the agents in the system.
- *Individual Rationality*: an allocation scheme is individually rational if agents are willing to participate in the scheme rather than opting out of it. It is commonly assumed that the utility of an agent choosing to opt out of a scheme, $\underline{u}_i(.)$, is 0. Hence, it is sufficient to ensure that the agents derive a utility $u_i \geq 0$ by being in the system.
- *Incentive Compatibility*: an incentive compatible system is one in which the agents will find no better option than to reveal their true type.

Amongst the class of mechanisms that satisfy the above properties, the VCG mechanism implements an efficient allocation under dominant strategies (i.e. each agent has a best strategy no matter what other agents' strategies are) (Mas-Colell et al., 1995). Using the VCG mechanism, our task allocation problem is then reduced to the following protocol which is shown in figure 6.1:

1. The centre receives the set of tasks $\boldsymbol{\tau}$ to be allocated from the agents (step 1).
2. The centre then posts these tasks in the vector $\boldsymbol{\tau}$ (step 3). Each agent i then reports its cost $\widehat{c}_i(K, \theta_i)$ (in the vector $\widehat{\boldsymbol{c}}(K, \boldsymbol{\theta})$) for completing a set of tasks in the set of allocations K along with the *reported* valuation $\widehat{v}_i(K, \theta_i)$ (in the vector $\widehat{\boldsymbol{v}}(K, \boldsymbol{\theta})$) it derives from having a set of tasks completed (step 4). In the rest of the

paper, we will superscript with '^' those variables and functions that are *reported* to the centre (auctioneer) to differentiate from those that are privately known. Of course, the reported values and costs can be different from the actual values and costs.

3. The centre then solves the following standard VCG auction equation (step 5):

$$\widehat{K}^* = \arg\max_{K \in \mathcal{K}} \sum_{i \in \mathcal{I}} [\widehat{v}_i(K, \theta_i) - \widehat{c}_i(K, \theta_i)] \tag{6.2}$$

and computes each transfer r_i in the vector $\boldsymbol{r}$ as:

$$r_i = \left[\sum_{j \in -i} \left[\widehat{v}_j(\widehat{K}^*, \theta_j) - \widehat{c}_j(\widehat{K}^*, \theta_j) \right] \right] - \left[\max_{K \in \mathcal{K}} \sum_{j \in -i} [\widehat{v}_j(K, \theta_j) - \widehat{c}_j(K, \theta_j)] \right] \tag{6.3}$$

where $-i \equiv \mathcal{I} \setminus i$.

4. The centre allocates the tasks according to the optimal allocation K^* and implements the transfers r_i (step 6).

The VCG mechanism results in an alignment of the goal of each agent with that of the mechanism designer via the use of the transfer part of the mechanism. Basically, each agent has as its best strategy the social optimum goal, which can only be achieved via a truthful revelation. That is, for each agent i,

$$\widehat{c}_i(K, \theta_i) = c_i(K, \theta_i) \tag{6.4}$$

and

$$\widehat{v}_i(K, \theta_i) = v_i(K, \theta_i) \tag{6.5}$$

Since the agents find it optimal to report their true valuations and costs, the centre thus finds the efficient allocation in step 3 (i.e. $\widehat{K}^* = K^*$). The second part of the transfer ensures that agents have $u_j \geq 0$ and thereby makes the mechanism incentive compatible.

We have thus presented a standard DRM for our task-allocation problem that achieves efficiency, incentive compatibility, and individual rationality under dominant strategy equilibrium. However, this mechanism only considers the cost and value of the tasks and disregards the uncertainty about the reliability of the agents in executing their tasks. Reducing this uncertainty through the use of the concept trust (as calculated in individual trust models) is one of our main goals (see chapter 1). To this end, in the next section we introduce trust as another dimension to be used in the computation of the efficient allocation and show why the standard VCG is neither incentive compatible nor efficient when trust is taken into account.

6.4 Trust-Based Mechanism Design

To incorporate trust, a further dimension needs to be added to the utility function in equation 6.1 which, in turn, requires both the allocation and payment schemes in the VCG mechanism to be modified to take this additional dimension into account. Having defined our mechanism (see section 6.5), we prove that it is incentive-compatible, efficient and individually-rational (in section 6.5.1). Before doing this, however, we first need to specify the generic properties that allow trust to be defined as a measure that can be used in computing efficient allocations.

6.4.1 Properties of the Trust Model

As we have seen in chapters 3 and 5, many computational trust models have been developed to allow agents to choose their most trustworthy interaction partners or negotiate with them (as discussed in section 6.2). However, at their most fundamental level, these models can be viewed as alternative approaches for achieving the following properties[2]:

1. As can be deduced from our discussion in chapter 3, the trust measure of an agent i in an agent j normally depends both on i's perception of j's POS and on the perception of other agents on j's POS. This latter point encapsulates the concept of *reputation* whereby the society of agents generally attributes some characteristic to one of its members by aggregating some/all the opinions of its other members about that member (see section 3.1.2, chapter 3). Thus, each agent considers this societal view on other members when building up its own measure of trust in its counterparts (Dasgupta, 1998). The trust of agent i in its counterpart j, $t_i^j \in [0, 1]$, is given by a function, $g : [0, 1]^{|\mathcal{I}|} \rightarrow [0, 1]$, (which, in the simplest case, is a weighted sum) of all POS measures sent by other agents to agent i about agent j as shown below:
$$t_i^j = g(\{\eta_1^j, \ldots, \eta_i^j, \ldots, \eta_N^j\}) \tag{6.6}$$
where $\eta_i^j \in [0, 1]$ is the POS of agent j as perceived by agent i and g is the function that combines both personal measures of POS and other agents' measures. In general, trust models compute the POS measures over multiple interactions. Thus, as in CREDIT, the level of success recorded in each interaction is normally averaged to give a representative value (see chapter 3 for a wider discussion). In our model, we use such a basic model whereby each agent records the success of a task and averages that with its past impressions (in CREDIT we deduce a model

[2]Note that we do not focus on a particular trust model. This is because trust models implement the above properties in their own ways and in different contexts. Therefore, we concentrate on these abstract properties to keep the focus on the relationship between trust and the design of an efficient mechanism. In so doing, we ensure that the properties of our mechanism are independent of any specific trust model.

of the agents' POS through a normal distribution and use a representative value from the confidence interval of that distribution).

2. Trust results from an analysis of an agent's POS in performing a given task. The more successful, the more trustworthy the agent is. Thus, the models assume that trust is monotonic increasing with POS. Therefore, the relationship between trust and POS is expressed as: $\frac{\partial t_i^j}{\partial \eta_i^j} > 0$, where t_i^j is the trust of i in agent j and η_i^j is the actual POS of agent j as perceived by i. Though it may seem that this property is quite rational (in that one does not reward bad behaviour with more trust), as we have seen in section 5.5, some models *do not* implement such a property but these are fairly rare (see discussion on Witkowski et al.'s (Witkowski et al., 2001) in section 5.5, chapter 5)).

Given the above, agents can update their trust rating for another agent each time they interact (both by recording their view of the success of their counterpart and by gathering new reports from other agents about it). Thus, if an agent's POS does not change, the trust measure in it should become more precise as more observations are made and received from other agents. Moreover, having the trust monotonic increasing with POS ensures Mirrlees's condition regarding fixed points in allocation schemes (Mirrlees, 1971) (which is a necessary condition for the mechanism to be efficient) is satisfied.

6.4.2 Augmenting the Task Allocation Scenario

In this section we show how trust is to be calculated and taken into account in the task allocation example we described in section 6.3. Here, any trust model satisfying the properties discussed in section 6.4.1 (such as CREDIT) can be used when actually building the system. The following changes are made (as shown in figure 6.2):

- Each agent i reports to the centre their POS vector:

  $$\widehat{\boldsymbol{\eta}}_{\boldsymbol{i}} = [\widehat{\eta}_i^1 \dots \widehat{\eta}_i^I]$$

  (step 1). This is the POS that an agent has observed about the other agents. This vector may not be complete if agents have not experienced any past interactions with other agents. However, this does not affect the properties of the mechanism since the centre will only pick those POSs that are relevant (and calculate trust according to these).

- The agents must also submit their respective trust calculation function (equation 6.6) that applies over the vector of all (or part of) other agents' reported POSs (i.e. $\widehat{\boldsymbol{\eta}}$), $\boldsymbol{t_i} = g(\widehat{\boldsymbol{\eta}})$, to the centre before the allocation of tasks (step 2). This allows the centre to compute the trust of agent i in all other agents (given i's own perception,

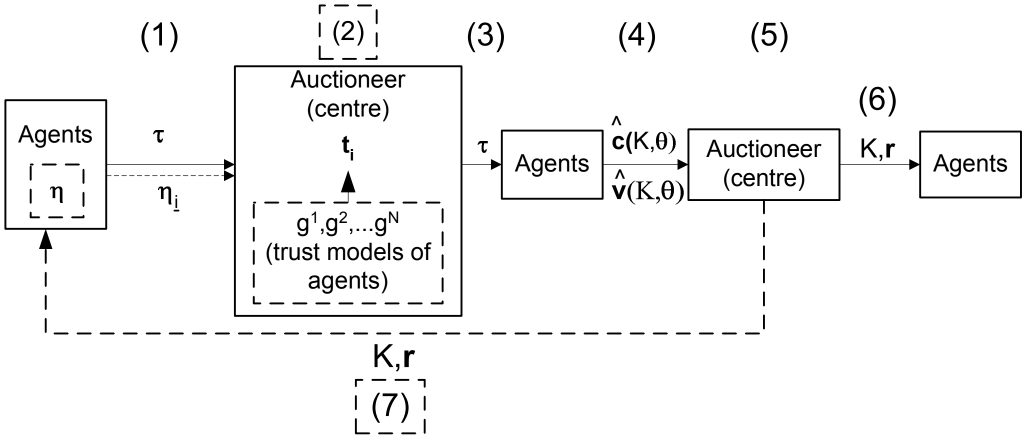


FIGURE 6.2: Simple task allocation using TBM. The dotted lines represent the modifications we make to the mechanism when using trust in the feedback loop. The $g^{\mathcal{I}}$ functions represent the trust functions that are used to aggregate all POS values from other agents into a common measure of trust.

as well as other agents' perceptions of the task performer's POS). Given that the trust t_i only affects the allocation of tasks originating from agent i, the latter has no incentive to lie about its trust function to the centre (otherwise it could result in i's task not being allocated to the agent deemed most trustworthy by i).

The trust function $g(.)$ may assign different weights to the reports of different agents depending on the level of similarity between the types of agents i and $-i$ (where $-i \equiv \mathcal{I} \backslash i$). Thus, given the trust functions and reports of POS of each agent, we now require the centre to maximise the overall *expected* valuation of the allocation (in step 5), as opposed to the valuation of the allocation independent of trust (i.e. which the standard VCG does). This is because an agent has a certain probability of completing the task to a degree of success which may be less than one. We denote as γ the *completion vector* of an allocation K which measures the level to which each task in an allocation is deemed completed. Thus, the expected value of an allocation is then $\Big(E_{[\gamma|K,\boldsymbol{t_i}]}\left[\sum_{i\in\mathcal{I}}\widehat{v}_i(K,\theta_i)\right] - \sum_{i\in\mathcal{I}}\widehat{c}_i(K,\theta_i)\Big)$ given the trust vector $\boldsymbol{t_i}$. This captures the fact that the agent i, that allocated the task, determines the value of γ. Moreover, agent j, to which the task has been allocated, incurs a cost independent of how agent i evaluates the task. This effectively means that the valuations are non-deterministic while the costs are deterministic. The centre thus determines the efficient allocation K^* (step 7) such that the value of the efficient allocation is maximised.

Having shown how to fit trust into the process of determining the value of allocations, in the next subsection we provide a simple example to show why the standard VCG solution of section 6.3 is not incentive compatible (and thus not efficient). This then motivates the search for a mechanism that is.

TABLE 6.1: A set of four agents in which agent 4 has proposed a task.

Agent i	c_i	η_i^1	η_i^2	η_i^3	t_4^i
1	40	0.4	1.0	0.8	0.5
2	80	0.6	1.0	0.8	1.0
3	50	0.5	1.0	0.9	0.86
4	∞	0.525	1.0	0.95	na

6.4.3 Failure of the VCG Solution

Consider a system of four agents where agent 4 has asked for a task τ to be allocated and its valuation of this task is $v_4(\tau, \theta_4) = 210$. Each agent i has a cost c_i to perform the task proposed by 4 (agent 4 has infinite cost to perform the task by itself) and does not derive any value from the task being performed. Now, suppose that the trust function of agent 4 is a weighed sum of the POS reports by the agents (i.e. $t_4^i = \alpha \,.\, \widehat{\boldsymbol{\eta}}^{\boldsymbol{i}}$ where $\alpha = [0.3 \quad 0.2 \quad 0.1 \quad 0.4]$). Note that we do not concern ourselves with the reports η_i^4 since the task is proposed by agent 4 itself. Table 6.1 shows the cost c_i of attempting the task, and the observed POS value of each agent, $\boldsymbol{\eta_i}$, as well as the trust computed by agent 4, t_4^i, if each agent reports truthfully on its $\boldsymbol{\eta_i}$.

The VCG solution of section 6.3 determines the allocation and payments based only on cost and valuations. However, this would clearly fail to find an efficient allocation since agent 1 would be allocated the task despite being the least trusted and hence most likely to fail. If we instead implemented the VCG mechanism with the *expected* valuations (taking into account the trust and POS reports), we then have $K^* = [0010]$ (i.e agent 3 is allocated the task), $r_1 = r_2 = 0$ and $r_3 = 210\gamma - 130$. Thus, agent 3 will then derive an average payment of $0.87 \times 210 - 130 = 52.7$. However, this scheme is not incentive-compatible because agent 2 can lie about η_2^3 by reporting $\widehat{\eta}_2^3 \leq 0.7357$ which will then lead to agent 2 being allocated the task and deriving a positive utility from this allocation. Note that this scheme is exactly that of (Porter et al., 2002) for a single-task scenario (with the modification that we use γ as a level of success rather than a binary indicator function of success or failure).

As can be seen, the VCG mechanism needs to be extended to circumvent this problem. Specifically, we require a mechanism that is efficient given the reports of the agents on their costs and valuations of allocations, as well as their observed POS vector (since the VCG is affected by false reports of POS). In effect, we need to change the payment scheme so as to make the truthful-reporting of POSs an optimal strategy for the agent again. Once this is achieved, the centre can then choose the efficient allocation based on expected utilities. The difficulty with designing such a mechanism is that the centre cannot check on the validity of POS reports of agents because it is based on a private observation carried out by the agent. Thus two agents may legitimately differ in their observed POS of another agent due to their different interaction histories with that agent.

6.5 The Trust-Based Mechanism

Before presenting our trust-based mechanism (TBM), we first introduce some new notation. Let the sum of utilities of all agents in a system given an allocation K and a completion vector γ be denoted as $U(K, \boldsymbol{\theta}, \gamma) = \sum_{i \in \mathcal{I}} v_i(K, \theta_i, \gamma) - \sum_{i \in \mathcal{I}} c_i(K, \theta_i)$. Then the expected utility $\overline{U}(K, \boldsymbol{\theta}, \gamma)$ before the allocation is carried out is $E_{[\gamma|K, \boldsymbol{t_i}]}\left[U(K, \boldsymbol{\theta}, \gamma)\right]$ where $\boldsymbol{\theta}$ is the vector containing all agent types. We also denote the marginal contribution of the agent i to the system given an efficient allocation $\widehat{K}^*$ as $mc_i = \overline{U}_{-i}(\widehat{K}^*, \boldsymbol{\theta}, \gamma) - \max_{K \in \mathcal{K}}\left[\overline{U}_{-i}(K, \boldsymbol{\theta_{-i}}, \gamma)\right]$ where $\max_{K \in \mathcal{K}}\left[\overline{U}_{-i}(K, \boldsymbol{\theta_{-i}}, \gamma)\right]$ is the overall expected utility of the efficient allocation that would have resulted if agent i were not present in the system. Now, we can detail TBM:

1. Find the efficient allocation $\widehat{K}^*$ such that:

$$\widehat{K}^* = \arg\max_{K \in \mathcal{K}} \overline{U}(K, \boldsymbol{\theta}, \gamma) \tag{6.7}$$

   This finds the best allocation; that is, the one that maximises the sum of *expected utilities* of the agents, conditional on the reports of the agents. We note here that we do not take into consideration the reward functions of the agents when calculating the overall utility since these rewards are from one agent to another and therefore do not make a difference when calculating the overall utility of the agents.

2. We now calculate the efficient allocation that would have resulted if an agent i's report taken out:

$$K^*_{-i} = \arg\max_{K \in \mathcal{K}} E_{[\gamma|K, \boldsymbol{t'_i}]}\left[U(K, \boldsymbol{\theta}, \gamma)\right] \tag{6.8}$$

   , where $\boldsymbol{t'_i} = g(\widehat{\boldsymbol{\eta}} \setminus \widehat{\boldsymbol{\eta}}_{\boldsymbol{i}})$. This computes how $\widehat{\boldsymbol{\eta}}_{\boldsymbol{i}}$ affects which allocation is deemed efficient.

3. We now find the effect that an agent's $\widehat{\boldsymbol{\eta}}_{\boldsymbol{i}}$ has had on its marginal contribution. Thus, find

$$D_i = \overline{U}(\widehat{K}^*, .) - \overline{U}(K^*_{-i}, .) \tag{6.9}$$

   This distils the effect of an agent's $\widehat{\boldsymbol{\eta}}_{\boldsymbol{i}}$ reports.

4. Given K^*, the payment r_i made to the agent i is then:

$$r_i = mc_i - D_i \tag{6.10}$$

   Naturally, if r_i is negative it implies that i makes a payment to the centre. The first part of the payment scheme, mc_i, calculates the effect that an agent's *presence* has had on overall expected utility of the system. We also subtract D_i to take into account the effect that an agent's POS report has on the chosen allocation. This is in line with the intuition behind VCG mechanisms in which an agent's report affects the allocation but not the payment it receives or gives.

We will now prove each of the properties of TBM in turn whilst intuitively explaining why the mechanism has the aforementioned properties.

6.5.1 Properties of our Trust Based Mechanism

Given the mechanism presented in the previous section, we now prove its main properties in the following order: incentive compatibility, efficiency, and individual rationality.

Proposition 1. TBM is incentive-compatible in ex-ante Nash Equilibrium.

Proof. We first need to calculate the expected utility, $E_{[\gamma|K,\boldsymbol{t_i}]}\left[u_i(K,\theta_i,\gamma)\right]$, that an agent derives from TBM because the goal of a rational agent is to maximise its expected utility. We note here that we are assuming that the agent is myopic in that it is only concerned with its current expected utility given the cost vector, $\boldsymbol{c}(K,\boldsymbol{\theta})$, the value vector, $\boldsymbol{v}(K,\boldsymbol{\theta})$, and the trust vector $\boldsymbol{t}$. The expected utility that an agent, $\overline{u}_i(\widehat{K}^*,\theta_i,\gamma)$, derives from an efficient allocation, as calculated from equation 6.7, given the reports of all agents in the system is:

$$\begin{aligned}\overline{u}_i(\widehat{K}^*,\theta_i,\gamma) &= E_{[\gamma|\widehat{K}^*,\boldsymbol{t_i}]}\left[v_i(\widehat{K}^*,\theta_i,\gamma)\right] - c_i(\widehat{K}^*,\theta_i) \\ &\quad + mc_i(\widehat{K}^*,\theta_i,\gamma) - D_i \\ &\quad = E_{[\gamma|\widehat{K}^*,\boldsymbol{t_i}]}\left[v_i(\widehat{K}^*,\theta_i,\gamma) - \widehat{v}_i(\widehat{K}^*,\theta_i,\gamma)\right] \\ &\quad - \Big(c_i(\widehat{K}^*,\theta_i) - \widehat{c}_i(\widehat{K}^*,\theta_i)\Big) + \\ &\overline{U}(K^*_{-i},\boldsymbol{\theta},\gamma) - \max_{K\in\mathcal{K}}\left[\overline{U}_{-i}(K,\boldsymbol{\theta_{-i}},\gamma)\right] \end{aligned} \tag{6.11}$$

¿From 6.11 we will firstly prove the following lemma:

Lemma 1. An agent has an equilibrium strategy to reveal its observed POS values.

Proof. We consider how $\widehat{\boldsymbol{\eta}}_{\boldsymbol{i}}$ affects $\overline{u}_i(\widehat{K}^*,\theta_i,\gamma)$. From equation 6.11 we observe that $\widehat{\boldsymbol{\eta}}_{\boldsymbol{i}}$ cannot affect $\overline{U}(K_{-i},\boldsymbol{\theta},\gamma) - \max_{K\in\mathcal{K}}\left[\overline{U}_{-i}(K,\boldsymbol{\theta_{-i}},\gamma)\right]$. Thus, an agent only has an incentive to lie so that $\widehat{K}^*$ is selected such that:

$$E_{[\gamma|\widehat{K}^*,\boldsymbol{t_i}]}\left[v_i(\widehat{K}^*,\theta_i,\gamma) - \widehat{v}_i(\widehat{K}^*,\theta_i,\gamma)\right] - \Big(c_i(\widehat{K}^*,\theta_i) - \widehat{c}_i(\widehat{K}^*,\theta_i)\Big)$$

is maximised. If an agent reveals its cost and valuation truthfully i.e. $\widehat{v}(.) = v(.)$ and $\widehat{c}(.) = c$, we then have the term as zero. Then an agent cannot gain from an untruthful reporting of $\widehat{\boldsymbol{\eta}}_{\boldsymbol{i}}$. If however, an agent is to gain from such an untruthful reporting, it needs to set either $\widehat{v}(.) < v(.)$ and $\widehat{c}(.) > c$ or both. However, doing so would decrease the chance of i successfully allocating a task or winning an allocation. Therefore, i would not reveal untruthful values for $\widehat{c}(.)$ and $\widehat{v}(.)$. Moreover, i will actually report truthfully its $\widehat{\boldsymbol{\eta}}_{\boldsymbol{i}}$ since this allows the centre to choose those agents that i deems to have a high POS (as well as helping other agents choose i as having a perception close to theirs). Thus, reporting $\widehat{\boldsymbol{\eta}}_{\boldsymbol{i}} = \boldsymbol{\eta}_{\boldsymbol{i}}$ is an ex-ante Nash equilibrium strategy. □

Given lemma 1, we can now show that TBM is incentive compatible. Suppose an agent is truthful about $\widehat{v}(.)$ and $\widehat{c}(.)$. Then it derives as utility:

$$\overline{U}(K^*_{-i}, \boldsymbol{\theta}, \gamma) - \max_{K \in \mathcal{K}} \left[\overline{U}_{-i}(K, \boldsymbol{\theta}_{\boldsymbol{-i}}, \gamma)\right]$$

Now assume that the agent lies about $\widehat{v}(.)$ and $\widehat{c}(.)$ so as to increase its utility. This then means that:

$$E_{[\gamma|\widehat{K}^*, \boldsymbol{t_i}]}\left[v_i(\widehat{K}^*, \theta_i, \gamma) - \widehat{v}_i(\widehat{K}^*, \theta_i, \gamma)\right] - \Big(c_i(\widehat{K}^*, \theta_i) - \widehat{c}_i(\widehat{K}^*, \theta_i)\Big) + \overline{U}(K'_{-i}, \boldsymbol{\theta}, \gamma) > \overline{U}(K^*_{-i}, \boldsymbol{\theta}, \gamma) \tag{6.12}$$

where K'_{-i} is the efficient allocation found with $\widehat{c}(.)$ and $\widehat{v}(.)$ without the report of $\boldsymbol{\eta}_{\boldsymbol{i}}$. However, as argued earlier, an agent would not report a lower value or a higher cost. Thus

$$E_{[\gamma|K, \boldsymbol{t_i}]}\left[v_i(\widehat{K}^*, \theta_i, \gamma) - \widehat{v}_i(\widehat{K}^*, \theta_i, \gamma)\right] - \Big(c_i(\widehat{K}^*, \theta_i) - \widehat{c}_i(\widehat{K}^*, \theta_i)\Big) \leq 0 \tag{6.13}$$

Furthermore, by the maximisation of step 2 of TBM:

$$\overline{U}(K'_{-i}, \boldsymbol{\theta}, \gamma) < \overline{U}(K^*_{-i}, \boldsymbol{\theta}, \gamma) \tag{6.14}$$

if all other agents report truthfully. Thus, TBM is incentive-compatible in a Nash equilibrium. □

Proposition 2. TBM is efficient.

Proof. Given that the agents are incentivised to report truthfully (proposition 1), the centre will calculate the efficient allocation according to equation 6.7 (i.e. $\widehat{K}^* = K^*$). □

Proposition 3. TBM is individually-rational (in expected utility).

Proof. We need to show that the expected utility of any agent from an efficient allocation K^* is greater than if the agent were not in the scheme (i.e. $\overline{u}_i(K^*, \theta_i, \gamma) \geq 0$). As a result of the inherent uncertainty in the completion of tasks, we cannot guarantee that the mechanism will be ex-post individually-rational for an agent. Rather, we prove that the mechanism is individually-rational for an agent if we consider expected utility.

Given truthful reports, the utility of an agent from equation 6.11 is $\overline{U}(K^*_{-i}, \boldsymbol{\theta}, \gamma) - \max_{K \in \mathcal{K}} \left[\overline{U}_{-i}(K, \boldsymbol{\theta_{-i}}, \gamma)\right]$. The first maximisation is carried out without the reports $\boldsymbol{\eta_i^{-i}}$,whereas the second maximisation is carried out over the set of agents $\mathcal{I} \setminus i$. Thus, the second maximisation is carried out over a smaller set than the first one. As a result:

$$\max_{K \in \mathcal{K}} \left[\overline{U}_{-i}(K, \boldsymbol{\theta_{-i}}, \gamma)\right] \geq \overline{U}(K^*_{-i}, \boldsymbol{\theta}, \gamma) \tag{6.15}$$

such that $\overline{u}_i(K^*, \theta_i, \gamma) \geq 0$. □

6.5.2 Instances of TBM

TBM can be viewed as a generalised version of the VCG mechanism in which there exist uncertainties about whether a set of agents will carry out an allocation and about the relevance of reports of POS by agents. In this section, we demonstrate its generality by analysing two specific instances of the mechanism.

6.5.3 Self-POS Reports Only

The non-combinatorial mechanism developed in (Porter et al., 2002) is a special case of TBM. Specifically, agents only report on their own POS (i.e. $\widehat{\boldsymbol{\eta}}_{\boldsymbol{i}} = \widehat{\eta}_i^i$) and agents assign a relevance of 1 to reports by all other agents. However, since in their model their is no notion of varying perceptions of success, we need to introduce the notion of a *report agent* that has $v(K, .) = 0$ and $c(K, .) = \infty$. This acts as a proxy to agents reporting the ex-post POS to the centre. This also caters for the problem of single POS reports as there is then no measure of t_i^j once j's report is removed (and hence $\overline{U}(K^*_{-i}, .)$ is undefined). The centre then calculates the efficient allocation as:

$$K^* = \arg\max_{K \in \mathcal{K}} \left[\overline{U}(\widehat{K}^*, \boldsymbol{\theta}, \gamma)\right] \tag{6.16}$$

and the payment to the agent i is $r_i = mc_i - D_i = mc_i$. The term $D_i = 0$ since, as a result of the report agent, $\overline{U}(\widehat{K}^*, .) = \overline{U}(K^*_{-i}, .)$ (because $\boldsymbol{t}$ is equal in both cases are the same).

6.5.4 Single-Task Scenario

Consider the single task scenario (as presented in table 6.1) where an agent k proposes a single task τ_k. Using equation 6.7, the efficient allocation is then simplified to:

$$K^* = \arg\max_{K \in \mathcal{K}} \left[\overline{U}(\widehat{K}^*, \boldsymbol{\theta}, \gamma)\right]$$

The payment to agent i, from equation 6.10, is then:

$$
\begin{aligned}
r_i = E_{[\gamma|K^*_{-i},\boldsymbol{t_k}]}\left[v_k(K^*_{-i},\theta_k,\gamma)\right] + \widehat{c}_i(\widehat{K}^*,\theta_i,\gamma) - \textstyle\sum_{i\in\mathcal{I}}\widehat{c}_i(K^*_{-i},\theta_i,\gamma) \\
-\max_{K\in\mathcal{K}}\left[E_{[\gamma|K,\boldsymbol{t_k}]}\left[v_k(K,\theta_k,\gamma)\right] - \textstyle\sum_{j\in -i}\widehat{c}_j(K,\boldsymbol{\theta_{-i}},\gamma)\right]
\end{aligned}
\tag{6.17}
$$

Since the above single-task scenario is an instance of the TBM, it is still incentive compatible. Therefore, when applying the above allocation scheme to the example, we can take the reported values of the agents as being truthful. Given this, the efficient allocation is agent 3 getting to do the task. Then, we need to check whether agent 3's report has made itself more attractive. To do so, we remove the report of agent 3 and end up with agent 4 having a trust vector $\boldsymbol{t_4^i} = [0.5 \quad 1.0 \quad 0.9]$ which again leads to agent 3 being allocated the task. Thus agent 3 will get an expected utility of $210*0.8667 - \ 50+\ 50\ -130-50 =\ 2$. Agent 1 and 2 no longer have an incentive to lie about the POSs since this would not increase their utility. However, suppose that, after the allocation, every type becomes common knowledge. Then agent 2 can deduce that lying about its costs *and* reported POS would allow its utility to increase. This would have been maximised when agent 2 reports $\widehat{c}_2(.) = 110$ and $\widehat{\eta}_2^3 = 0$. However, before the allocation is carried out and payments are made, agent 2 would not know about the private types of other agents and may reduce its chance of deriving a positive utility by reporting $\widehat{c}_2(.) > c_2(.)$. Furthermore, agent 2 does not report $\widehat{\boldsymbol{\eta}}_{\boldsymbol{2}}^{\boldsymbol{-2}} < \boldsymbol{\eta_2^{-2}}$ since then $u_2(.) = 0$ even if it wins the allocation. A similar argument applies to agent 1. Thus, the mechanism has an ex-ante Nash Equilibrium of truthful reporting.

6.6 Experimental Evaluation

Given that our TBM relies on the agents' individual trust model, it is important to show how the trust models can affect the (efficient) outcome chosen. This is because, trust models at the individual level need a number of interactions to refine their measures and may also be affected by biased reports (see discussion in sections 3.1.2.2 and 3.2.2 in chapter 3) such that the actual efficient outcome (i.e. one which maximises the utility *and* chooses the most reliable agents) may, at times, not be chosen. Hence we empirically evaluate TBM by comparing it with the fault tolerant mechanism (FTM) of (Porter et al., 2002) (this is chosen because it also deals with the POS of agents as discussed in sections 6.2 and 6.5.2) and the standard VCG. We refer to task performing agents as contractors in what follows. In our experiments we perform 500 successive allocations, in the scenario described in section 6.4, with six agents each given one task to complete. After each allocation, contractors perform tasks and the level of success is measured and reported to all agents. Each agent can then update its measure of the contractors' POSs as well as the contractors' trustworthiness as discussed in section 6.4.1. The valuations and POS of each agent are obtained from a uniform distribution and the costs are the same for all tasks. We iterate the process and average the results (here for 200 iterations). Given the properties of TBM and FTM we postulate the following hypotheses and validate them as shown below:

Hypothesis 6. TBM always chooses the efficient allocation (K^*) in the long run.

This hypothesis reflects the fact that we expect agents in TBM to take a number of interactions to model the true POS of their counterparts, using their individual trust models. After this time, however, the mechanism can choose those contractors that are most successful at completing a given task. As can be seen in figure 6.3, the optimal allocation chosen by TBM, K^*TBM, reaches the efficient allocation K^* (given *real* POSs) after 116 interactions.[3] After 116 interactions, the POS of each contractor is accurately modelled, as is the trust of agents in their contractors. Thus, the most trusted and utility maximising allocation is found by the TBM. This result is observed for all cases where the POSs of contractors are varied.

Hypothesis 7. TBM finds better allocations than FTM when contractors' own reported POS are biased.

While FTM only takes into account a contractor's own reports, TBM uses the trust model of the various individual agents (which take into account reports not only from the contractor) to make an allocation. In the particular trust model (based on CREDIT) we use in TBM, an agent can give different weights to reports from different agents (as shown in section 6.4.3). We therefore varied the weight w, assigned to a contractor's report of its own POS in the trust model of an agent. Here we exemplify the cases where $w = 0.5$ (i.e. the contractor's report is given equal weighting to the agent's perceived POS), $w = 0.25$ and $w = 0$ (i.e. no importance is given to the contractor's report).

As can be seen, our hypothesis is validated by the results given in figure 6.3 (with normalised expected values). Note here that K^*VCG is the allocation independent of POSs or if POSs of agents are all equal. We note as K^*TBM_w the allocation chosen by TBM with a weight w.

In more detail, TBM_0 (i.e. TBM) reaches the optimal allocation K^* (i.e. equivalent to zero bias from the seller) after 116 iterations, while $TBM_{0.25}$ and $TBM_{0.5}$ settle around a sub-optimal allocation (the expected value of which decreases with increasing w). Moreover, FTM is seen to settle at $K^*FTM = 0.8$ after 82 iterations. In general, it is noted that FTM always settles at $K^*FTM < K^*$ (and sometimes even $K^*FTM < K^*VCG$ as in figure 6.3 depending on the valuations agents have for the tasks). This result is explained by the fact that the biased reports cause biased trust values to be obtained by the centre which then chooses a sub-optimal allocation (i.e. less than K^* which chooses agents according to their 'real POSs'). $TBM_{0.25}$ and $TBM_{0.5}$ are less

[3]The results were validated using a student's t-test with two samples of 100 and 200 iterations assuming equal variances with means $\mu_1 = 0.99999$ and $\mu_2 = 1.0$ and p-value $p = 0.778528$. This means that the difference between the means is not significant.

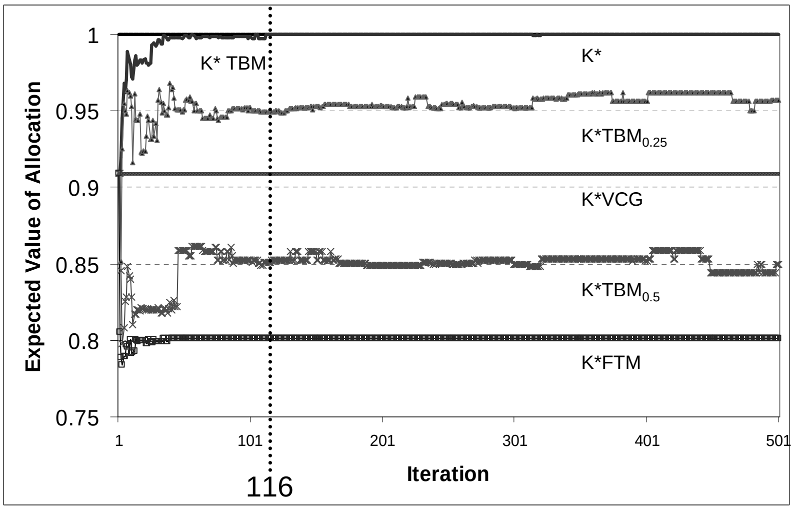


Figure 6.3: Expected value of chosen allocations for TBM and FTM where $K^* = 1$, $K^*VCG = 0.909$, and at equilibrium, $K^*TBM = 1$, $0.97 > K^*TBM_{0.25} > 0.94$, $0.86 > K^*TBM_{0.5} > 0.84$, and $K^*FTM = 0.8$.

affected by biased reports since the weighted trust model reduces the effect of bias on the overall trust values (but still affects the mechanism). In most trust models, however, $w \geq 0.5$ is never given to the contractors' POS report and here it only represents an extreme case (Ramchurn et al., 2004b). Moreover, if the bias is removed, then FTM and the weighted TBMs behave the same as TBM since the agents then perceive the same POS and all achieve K^*. It was also observed that the speed with which TBM and FTM achieve K^* also depends on the difference between the optimum allocation and other allocations. This is because the smaller the differences, the harder it becomes to differentiate these allocations given imperfect estimations of POSs (i.e. the larger the samples, the more accurate the POSs are, hence the longer the learning rate).

6.7 Summary

This chapter concludes our work on trust in general. At the individual level, we first developed the CREDIT trust model in chapter 5 and showed how it could be used by agents in direct negotiations when the agreements reached are prone to uncertainty. In this chapter, at the system level, we have developed a trust based mechanism that takes the burden of computation from the agent in order to come to optimal allocations while being robust to uncertainty. In particular, we have introduced the notion of Trust-Based Mechanism Design as a generalisation of the VCG mechanism by using the trust model of individual agents in order to generate efficient allocations. We have developed a Trust-Based Mechanism and proved that it is efficient, individually rational, and incentive compatible. Moreover, we have empirically evaluated TBM and shown that it *always* achieves the optimum allocation in the long run and achieves better allocations than its closest comparison when contractors provide biased reports of their probability of sucess

(POS).

Generally speaking, through CREDIT and TBMD, we have achieved our main objectives (set in chapter 1) with regards to reducing uncertainty about the reliability and honesty of agents through an agent's reasoning mechanism (i.e. through CREDIT) and through the protocol (i.e. through TBMD). However, these models do not specifically consider uncertainties about the action set and preferences of the agents. In the next chapter, we complement this with a novel PN mechanism that can be used in bargaining encounters to reduce such uncertainties through the use of arguments. These arguments aim, on the one hand, to better explore the preferences of the bargaining agents and, on the other, to reduce the space of offers (i.e. the action set) agents need to search to find an agreement.

Chapter 7

Persuasive Negotiation for Autonomous Agents

In chapters 5 and 6, we presented CREDIT and TBMD as models that reduce the uncertainty that arises in negotiations between autonomous agents. While reducing uncertainty about the reliability or honesty of agents, these models do not consider uncertainty about the action set and the preferences of agents in bargaining encounters. However, as stated in chapter 1, techniques that can reduce these uncertainties can help ensure that agents are able to reach better agreements faster. To this end, in this chapter we develop a new model of PN that attempts to reduce such uncertainties through the use of persuasive arguments. These arguments are rewards that are either given or asked from one agent to another during the bargaining process. In our model, a reward implies a constraint on the outcomes of future encounters (in favour of one agent or the other) such that the expected outcome under this constraint entices an opponent to accept an offer in the present encounter. Our model of PN consists of both the protocol and the reasoning mechanism that allows agents to exchange such arguments.

In more detail, we build upon existing protocols (Sierra et al., 1998; Bentahar et al., 2004) (as discussed in section 2.3) to manage the commitments that arise during PN using dynamic logic. Given this, we develop a new algorithm that can be used with the offers generated by standard (non-persuasive) negotiation tactics so as to compute rewards. Thus, we empirically show that the agents generally reach agreements with higher expected utility when they use our algorithm than when they do not. Moreover, we develop a new reward-based tactic (RBT) for PN that aims to optimally determine offers and rewards during the negotiation. We empirically evaluate the model and show that our RBT is able to reach agreements with even higher expected utility than standard negotiation tactics (with or without the PN component). Finally we evaluate RBT when the properties of the negotiating agents are varied and therefore identify the main factors that impact on the efficiency and effectiveness of rewards.

The rest of this chapter is structured as follows. Section 7.2 details the theory we use to describe the protocol agents will use to negotiate. Section 7.3 describes the reasoning mechanism used by agents to specify and evaluate offers and counter offers as well as the rewards they may use in a negotiation scenario based on an MMPD. Then the system is experimentally evaluated in section 7.4. Section 7.5 finally summarises the main contributions of this work and their implications for practical applications. First, however, we discuss the main issues that arise when arguments are used in negotiation in the following section.

7.1 Introduction

As was discussed in chapters 1 and 2, bargaining between autonomous agents normally proceeds as a series of offers and counter offers (Fatima et al., 2004; Faratin et al., 1998). These offers generally indicate the preferred outcome for the proponent and the opponent may either accept or counter-offer a more viable outcome. Recently, there have been a number of attempts to enrich this negotiation process by allowing agents to express different preferences or information during negotiation. These preferences or information have been generally characterised as arguments (see chapter 2 for more details) which aim to support a particular offer and therefore help in persuading an opponent to accept it. These arguments can either contain some form of justification or represent some form of reward, threat, or appeal. These two mechanisms represent the two main ways of performing ABN, that is justification based negotiation and persuasive negotiation (see sections 1.4 and 2.2 respectively). In this work, we are mainly interested in PN (for reasons discussed in chapter 1) where the rewards or threats have a clearer impact on an agent's utility.

However, introducing threats or rewards in the negotiation process impacts on many aspects of an agent's reasoning mechanism and, to this end, various attempts at dealing with these were discussed in chapter 2. Despite these works, however, much work still remains to make these approaches provably better than non-PN negotiation mechanisms (e.g., Faratin et al. (1998); Fatima et al. (2004)). In more detail, even though threats or rewards imply either a reduction of or an increase in an agent's utility respectively, these types of arguments have never been given clear semantics in terms of the actions or events that can be properly assessed to evaluate their impact. This is particularly important when it comes to implementing such mechanisms and comparing their efficiency with other negotiation mechanisms. Moreover, the use of threats raises the issue of non-credible alternatives (or empty threats) (Hovi, 1998). Indeed, if we take the case of two agents which do not have any information about their opponent's preferences, it is nearly impossible for them to make any credible threat, and there might be no purpose to enact threats if it costs an agent to do so (see (Hovi, 1998) for more details). In addition to this, the literature on negotiation has shown that, while threats can allow good agreements to be reached in specific settings, it is normally not recommended to use them since they usually cause mutual recriminations and the breakdown of relationships (repeated encounters) (Raiffa, 1982; Fisher and Ury, 1983; Schelling, 1963).

Given this background, we focus on the use of rewards in our model of persuasive negotiation in repeated encounters. Rewards have the advantage of having a clear economic benefit for the agent receiving it and entail a direct commitment by the agent giving it to continue a long term relationship which is beneficial to both participating agents (as opposed to threats which break relationships down and are not guaranteed to be enforced which makes them harder to assess in a negotiation encounter). This aspect of rewards makes it particularly suited to our objectives of reducing uncertainty over repeated encounters as in CREDIT and TBMD (see section 1.5 chapter 1). As discussed in section 2.2, rewards have mostly been pictured as promises to give a particular resource or prize at a later point in time. In our work, we propose that agents may also 'ask' for rewards. This is common in negotiations where the negotiators ask for a favour in future for accepting to concede in the current round of negotiation (Raiffa, 1982; Fisher and Ury, 1983). The use of such rewards (given or asked) is, however, different from negotiating multiple issues at the same time (as shown in CREDIT) where the trade-off is normally made on the negotiated issues and the agreement is settled then and there (Fatima et al., 2002, 2004). In contrast, rewards are contingent upon acceptance of an offer and there is normally some uncertainty as to whether and to what extent they will be carried out. This uncertainty exists because agents may not want to clearly define the nature of the rewards since they do not know their future costs or the probability of meeting again (e.g., a seller may not give the full extent of a discount even if it promised to give it earlier, or a buyer may ask for a low price on a car with a promise to buy another similar car in future at the same price but the buyer may change her mind and buy another car at a lower price). Given this, a key issue that arises is that of determining what actually constitutes a reward in agent based negotiation. To resolve this issue, we allow rewards to be an endogenous factor affecting the negotiation (i.e. built in operations on the negotiation object) rather than leaving the notion of rewards as an exogenous aspect of negotiations (i.e. as an external object to be given or asked). In so doing, it is possible to define a general decision making model that evaluates rewards thus defined as well as those that are defined for other application contexts. Moreover, it is possible to analyse the properties of this mechanism with a relatively small number of constraints (i.e. the preferences and attributes of the agents) that are general enough to fit any given context. This allows us to formulate a better analysis of the problem and develop the persuasive negotiation mechanism that has a better grounding than most models which stand on abstract bases.

In the context of this thesis, we apply the persuasive negotiation to long-term relationships (i.e. repeated encounters) between autonomous agents and devise mechanisms to rectify the shortcomings we identified in other models in section 2.3.

- First, while making offers about short term agreements, agents are also allowed to offer or ask for rewards in the form of constraints over future agreements. In so doing, we also tackle the problem of bargaining in long-term relationships (Muthoo, 1999), an aspect that has been overlooked by the agent's community.

- Second, we provide a novel protocol that specifies commitments that agents make to each other in engaging in persuasive negotiations. Specifically, the protocol aims to show how commitments can be made and retracted by issuing proposals and rewards or by performing some actions.

- Third, we provide a Reward Generation Mechanism (RWG) that specifies when and what arguments can be sent during a bargain. Moreover, we show these arguments should be calculated relative to a given offer as calculated by non-PN negotiation tactics.

- Fourth, we devise a novel reward based tactic (Reward Based Tactic (RBT)) for generating offers and arguments and show that it can lead to better outcomes than non-PN tactics (tagged with RWG and without).

- Fifth, we analyse the properties of our RBT under different conditions in order to deduce which are the most important factors that affect the effectiveness and applicability of arguments in bargaining.

7.2 The Negotiation Protocol

Negotiation proceeds via an exchange of offers and counter offers. In general, the specification of such a protocol is rather simple in that there is only one type of commitment (see definition 1.4) upheld by each agent at any one time (that is enacting the proposal if its offer is accepted). However, extending the protocol to encapsulate arguments means that other commitments (pertaining to the enactment of the content of arguments) must be specified for the agents issuing these arguments (Walton and Krabbe, 1995; McBurney et al., 2003; Bentahar et al., 2004). These commitments can then be checked by an institution or arbitrator in order to make sure that the agents are doing what they are supposed to and thus provide guarantees of proper behaviour (as discussed in section 1.2).

As discussed in section 2.1 there are a number of representations, such as Sierra et al.'s state machines or McBurney's commitment rules, that can be used to specify how these commitments can be made or retracted by the illocutions (what the agents say) and the actions (what the agents do). However, given that arguments are likely to result in a large number of states and state transitions and that the enactment of arguments requires clear semantics of actions to be performed, we specify our protocol in terms of Harel's dynamic logic (**DL**) (Harel, 1984). This type of action-based logic is particularly suitable for specifying programs or sets of actions which have start and termination conditions and constructs similar to a negotiation encounter. For example, as shown in section 2.1, the work of Bentahar et al. (2004) provides a particular characterisation of aguments using a combination of dynamic logic and CTL (computation tree logic). However, as we argue in section 2.3, their work does not deal with promises of future rewards as arguments and these have a particular meaning that is not captured by the commitments Bentahar et al.'s use. To remedy this, in our model we extend their work to cater for rewards that act as arguments. To this end, we first provide a brief overview of the constructs of dynamic logic and then specify the protocol in detail.

7.2.1 Background

The main components of **DL** are described as follows (see (Harel, 1984) for more details). Agents perform atomic actions $a_0, a_1, .. \in \Pi_0$. Π_0 represents the set of all atomic actions. Atomic programs are basic and indivisible; they execute in a single step. They are called atomic because they cannot be decomposed further. Γ is the set of formulae. The formulae in Γ are true or not in given states and the agents change states according to the actions they perform.

A program Π is generated from Π_0 by composing atomic actions using the following operators $;, ^*, ?, \cup$. $a_n; a_m$ signifies that a_m is performed after a_n (i.e. sequential composition) while a_n^* implies an iteration of a_n an indeterminate number of times, $\varphi?$ tests whether φ is satisfied in the current state, and $a_n \cup a_m$ specifies a non-deterministic execution of either a or b. Moreover, $[a]\varphi$ denotes that after program $a \in \Pi$ is executed, $\varphi \in \Gamma$ is *necessarily* true. $\langle a \rangle\varphi$ denotes that after program $a \in \Pi$ is executed, it is *possible* that $\varphi \in \Gamma$ is true. We also introduce the predicates *Do* to denote the action of making a formula true and *Done* to check whether an action has been executed (then true) or not (then false). Thus $[Do(\varphi)]\varphi$ means that after the execution of $Do(\varphi)$, φ is necessarily true. Similarly, $[a]Done(a)$ means that after executing a, $Done(a)$ is true. Finally $a\perp$ denotes that the execution of program $\overline{a}$ is not possible in any state. The propositional operators $\wedge, \vee, \neg, \leftrightarrow$, and 1 can be defined from $\rightarrow$ and 0 in the usual way.

In DL we first capture the set of all states of the world through the set S. Then, $\rho : \Pi_0 \rightarrow 2^{S \times S}$ is a function taking a program and giving the corresponding set of pairs of starting and end states. In our model, the states of the world are completely represented by the 'social' commitments ('social' since they result from a public expression of a commitment that can be tracked by everyone in the society). We denote social commitments with the predicate SC. Agents therefore make social commitments to each other about particular actions which may involve deals or contracts as defined in chapter 4 (the result of the negotiation encounter), noted as $(x_n = v_n) \wedge ... \wedge (x_m = v_m)$, agents come to during and after a negotiation dialogue respectively. Here $x_m = v_m$ means that an issue x_m in the deal takes a value v_m (we detail these in the next section).[1] The social commitments and enactment of deals are well formed formulae that can be made true or false according to the actions agents take. Hence, the function ρ in our model takes an action and returns formulae that represent the beginning and end state of that action. We will show the start and end states using the commitments and deals. Thus we give semantics to the negotiation dialogue in what follows.

7.2.2 The Syntax

Agents negotiate by sending illocutionary particles which contain offers and counter-offers. These illocutionary particles are considered to be actions as per speech-act theory (see section 1.4). Illocutions, from the set $I \subseteq \Pi_0$, generally talk about other illocutions (to be sent at a later time) or about contracts that can be made between the pair of negotiating agents. The set of contracts to be enacted by a group of agents $g \in G$ where $G \subseteq Ag$, is denoted as $\mathcal{O}_{\mathcal{G}}$. In more detail, we refine the definition of a contract from chapter 4 to mean a composition of a number of actions noted as $\langle Do(x_1 = v_1 \wedge x_2 = v_2 \wedge ... \wedge x_n = v_n)\rangle \in \mathcal{O}_{\{\alpha\}}$ which implies that agent α is to ensure that $(x = v) \in WFF$ (i.e. issue x takes the value v is a well formed formula) such that $[Do(x = v)](x = v)$. A contract would obviously contain some actions to be performed by the sender and some by the receiver (as in section 4.1) such that $O_{\{\alpha,\beta\}} = O_{\{\alpha\}} \cup O_{\{\beta\}}$. We require that each variable x in a deal occurs at most once and that the number of variables and the values taken by them is finite.

We conceive of two general classes of illocutions that can be used in persuasive negotiation. The first consists of negotiation illocutions I_{neg} that are used in negotiation, while the second contains those illocutions I_{pers} that are added to form the persuasive part of negotiation. Moreover, both these classes of illocutionary acts form the set $I = I_{nego} \cup I_{pers}$. In a dialogue between agents α and β for example, we note $I_\alpha^\beta \subseteq I$ as being those illocutions that are sent by α to β. Finally, the set I_α and I_β denote the set of all illocutions that α and β can send respectively. In what follows, we detail the syntax of each of these illocutions.

7.2.2.1 Negotiation Illocutions

I_{neg} is the set of the usual negotiation illocutions noted as $i(\alpha, \beta, p) \in I_{neg}$ where $i \in \{propose, accept\}$. These illocutions are described as follows:

- $propose(\alpha, \beta, p)$ — denotes that α sends a proposal to β to *accept* the deal given in $p \in \mathcal{O}_{\{\alpha,\beta\}}$.

[1]Other mathematical operations such as $\leq, =, \geq$ can also be used in contracts.

- *accept*(α, β, p) — denotes that α accepts to enact the contents of $p \in \mathcal{O}_{\{\alpha,\beta\}}$ that it is supposed to perform (i.e. the part $\mathcal{O}_{\{\alpha\}}$).

7.2.2.2 Persuasive Illocutions

We specify persuasive illocutions as follows: $i(\alpha, \beta, p, q) \in I_{pers}$ where $q \in Deals \cup I$ and $p \in \mathcal{O}_{\{\alpha,\beta\}}$ and $i \in \{askreward, reward\}$. As for negotiation illocutions, we specify below the type q takes in the illocution.

- *reward*(α, β, p, q) — denotes that α will reward β with $q \in \mathcal{O}_{\{\alpha\}} \cup I_\alpha$ if β accepts the deal proposed in $p \in \mathcal{O}_{\{\alpha,\beta\}}$. As can be seen, q can either be a deal that is favourable to β or an illocution that will help β in future (e.g. enhance the reputation of β or an accept of a deal to be presented at a later time).
- *askreward*(α, β, p, q) — denotes that α asks for a reward $q \in \mathcal{O}_{\{\beta\}} \cup I_\beta$ from β if β accepts the offer presented in $p \in \mathcal{O}_{\{\alpha,\beta\}}$.

Having exposed the syntax of these illocutions, we next describe the components that allow us to give semantics to these illocutions.

7.2.3 Semantics of Illocutions

As discussed in section 7.2.1, the actions or programs performed by agents result in changes in the state of the world. In our model, programs consist of a number of illocutionary acts or the execution of deals. To give semantics to our model we exploit the theory presented by Bentahar et al. (2004). In their model, the authors prescribe commitments that hold in different states of the world and agents are able to navigate between different states through the actions they perform. In short, these actions lead to some commitments becoming true or false (i.e. commitments are equivalent to well-formed formulae in our model). We therefore extend the work of Bentahar et al. to incorporate the notion of persuasive negotiation. To this end, we first conceive of $Comms$ as the set of social commitments that can be made in a dialogue as a result of illocutions being uttered and that can be retracted as other illocutions are uttered or other actions are executed. At the beginning of a negotiation dialogue (i.e. before any agent says anything), all the commitments are false. As the negotiation proceeds, some will become true (active) or false (inactive) according to the illocutions sent. Some commitments might also become false after some actions are performed after negotiation. In general, we specify a commitment in the following way:

$$SC(\alpha, \beta, \varphi, q) \in Comms$$

which implies a social commitment by α to β to commit to q if φ is true. By specifying φ in terms of the state resulting from the execution of an illocution or a contract, it is then possible to define different commitments that result from issuing *propose*, *accept*, *reward*, or *askreward* as we show in the following subsections.

7.2.3.1 Basic Axioms

We start with the basic axioms and explain each of them.

- $[propose(\alpha, \beta, p)]SC(\alpha, \beta, Done(accept(\beta, \alpha, p)), p)$. This means that α commits to enacting p if β accepts the proposal. We deal with semantics of *accept* in the next section.
- $\Big[reward(\alpha, \beta, p, q)\Big] \wedge \begin{array}{l} SC(\alpha, \beta, Done(accept(\beta, \alpha, p)), p) \\ SC(\alpha, \beta, Done(accept(\beta, \alpha, p); p), q) \end{array}$.

  This means that α commits to q and its part of the deal p if β accepts the deal p.[2]
- $\Big[askreward(\alpha, \beta, p, q)\Big] \wedge \begin{array}{l} SC(\alpha, \beta, Done(accept(\beta, \alpha, p)), p) \\ SC(\beta, \alpha, Done(accept(\beta, \alpha, p); p), q) \end{array}$. This means that β is committed to giving q to α if β ever accepts p and p is enacted at a later point in time. Moreover, α commits to enacting the proposal if β accepts.

We next outline the axioms that specify the constraints that exist over proposals and rewards:

- *Mutually exclusive proposals*

  $$\bigwedge_{\alpha,\beta,p,p',p'\neq p} SC(\alpha, \beta, Done(accept(\beta, \alpha, p)), p) \rightarrow \neg SC(\alpha, \beta, Done(accept(\beta, \alpha, p')), p')$$

  This is a basic statement saying that our protocol does not keep previous offers (here $p' \in \mathcal{O}_{\{\alpha,\beta\}}$) on the negotiation table.
- *Mutually exclusive rewards*

  $$\bigwedge_{\alpha,\beta,p} \left(\bigwedge_{q,q',q\neq q'} SC(\alpha, \beta, Done(acc; p), q) \rightarrow \neg SC(\alpha, \beta, Done(acc; p), q') \right)$$

  where $acc = accept(\beta, \alpha, p)$ if q or q' is a reward offered by α, $acc = accept(\alpha, \beta, p)$ if α asked for a reward q or q', and $q, q' \in \mathcal{O}_{\{\alpha,\beta\}} \cup I_\alpha$.

  This implies that there cannot be two rewards associated with the same deal at any given time.

[2] We assume here that the contract p is executed after the *accept*. In other cases, we might have intermediate contracts being enacted between the reception of the *accept* and the enactment of p but we do not consider this here.

- *Mutually exclusive rewards and proposals*

$$\bigwedge_{\alpha,\beta,q,q'} \left(\bigwedge_{p,p'} \neg(SC(\alpha,\beta,Done(accept(\beta,\alpha,p);p),q) \wedge SC(\beta,\alpha,Done(accept(\beta,\alpha,p');p'),q')) \right)$$

where $p, p' \in \mathcal{O}_{\{\alpha,\beta\}}$.

In essence this means that α's commitment to giving a reward q and enacting p cannot hold together with a commitment by β to give q' to α (which would have asked for it before hand) and enacting p'.

7.2.3.2 Dynamics of Commitments

Here we detail those axioms that illustrate the interaction between different commitment types.

Accepting Proposals and Rewards

- Accepting a proposal:

$$\begin{aligned} SC(\alpha,\beta,Done(accept(\beta,\alpha,p)),p) \rightarrow & [accept(\beta,\alpha,p)] \\ & \neg SC(\alpha,\beta,Done(accept(\beta,\alpha,p)),p) \\ & \wedge SC(\alpha,\beta,true,p') \wedge SC(\beta,\alpha,true,p'') \end{aligned}$$

where $p' \in \mathcal{O}_{\{\alpha\}}$, $p'' \in \mathcal{O}_{\{\beta\}}$, and $p = p' \cup p''$. Here we express that both agents are committed to enacting the content of the deal if the recipient of the offer accepts.

- Accepting a reward:

$$\begin{aligned} SC(\alpha,\beta,Done(accept(\beta,\alpha,p);p),q) \rightarrow & [accept(\beta,\alpha,p)] \\ & \neg SC(\alpha,\beta,Done(accept(\beta,\alpha,p);p),q) \\ & \wedge SC(\alpha,\beta,true,p') \wedge SC(\beta,\alpha,true,p'') \\ & \wedge SC(\alpha,\beta,Done(p),q) \end{aligned}$$

where $p' \in \mathcal{O}_{\{\alpha\}}$, $p'' \in \mathcal{O}_{\{\beta\}}$, and $p = p' \cup p''$. This signifies that α and β commit to enacting the proposal if the proposal is first accepted and α will give the reward $q \in \mathcal{O}_{\{\alpha\}} \cup I_\alpha$ if p is enacted.

- Accepting a request for a reward:

$$\begin{aligned} SC(\alpha,\beta,Done(accept(\alpha,\beta,p);p),q) \rightarrow & [accept(\beta,\alpha,p)] \\ & \neg SC(\alpha,\beta,Done(accept(\alpha,\beta,p);p),q) \\ & \wedge SC(\alpha,\beta,true,p') \wedge SC(\beta,\alpha,true,p'') \\ & \wedge SC(\beta,\alpha,Done(p),q) \end{aligned}$$

  where $p' \in \mathcal{O}_{\{\alpha\}}$, $p'' \in \mathcal{O}_{\{\beta\}}$, and $p = p' \cup p''$. This signifies that α and β commit to enacting the proposal if the proposal is first accepted and β will give the reward $q \in \mathcal{O}_{\{\beta\}} \cup I_\beta$ if the proposal p is enacted.

Changing Offers or Arguments

- A new proposal after another proposal:

$$\bigwedge_{\alpha,\beta,p,p',p\neq p'} \big(SC(\alpha,\beta,Done(accept(\beta,\alpha,p)),p) \rightarrow [propose(\alpha,\beta,p')]\varphi\big)$$

  where $SC(\alpha,\beta,Done(accept(\beta,\alpha,p')),p') \wedge \neg SC(\alpha,\beta,Done(accept(\beta,\alpha,p)),p) = \varphi$. Here we show how commitments to a previous proposal p are revoked when a new offer p' is made.

- A new proposal after a reward:

$$\bigwedge_{\alpha,\beta,p,p',p\neq p'} \big(SC(\alpha,\beta,Done(acc;p),q) \rightarrow [propose(\alpha,\beta,p')]\varphi\big)$$

  where $\varphi = SC(\alpha,\beta,Done(accept(\beta,\alpha,p')),p') \wedge \neg SC(\alpha,\beta,Done(acc;p),q)$, $acc = accept(\beta,\alpha,p)$ if α has offered reward q and $acc = accept(\alpha,\beta,p)$ iff β has asked for reward q. Here we specify that an agent revokes a commitment to p and reward q (given or asked for) if it proposes a new offer p'.

- A new reward following another reward:

$$\bigwedge_{\alpha,\beta,p,q,q',q\neq q'} \big(SC(\alpha,\beta,acc,q) \rightarrow [reward(\alpha,\beta,p,q')]\varphi\big)$$

  where $\varphi = SC(\alpha,\beta,Done(accept(\beta,\alpha,p);p),q') \wedge \neg SC(\alpha,\beta,Done(acc;p),q)$, and $acc = accept(\beta,\alpha,p)$ iff α offers reward q and q' and $acc = accept(\alpha,\beta,p)$ iff β asks for reward q or q'. This specifies that α commits to a new reward q' and decommits from a past reward q (given or asked) if it offers a new reward given the same proposal p.

- A new proposal and reward following another proposal and reward:

$$\bigwedge_{\alpha,\beta,q,p,p',p\neq p'} \big(SC(\alpha,\beta,Done(acc;p),q) \rightarrow [reward(\alpha,\beta,p',q)]\varphi\big)$$

  where $\varphi = SC(\alpha,\beta,Done(accept(\beta,\alpha,p');p'),q) \wedge \neg SC(\alpha,\beta,Done(acc;p),q)$, and $acc = accept(\beta,\alpha,p)$ iff α gives reward q and $acc = accept(\alpha,\beta,p)$ iff β asks for reward q. This says that α commits to a new proposal p' and the same reward q.

- Asking for a reward after another reward:

$$\bigwedge_{\alpha,\beta,p,q,q',q\neq q'} \big(SC(\alpha,\beta,Done(acc;p),q) \rightarrow [askreward(\alpha,\beta,p,q')]\varphi\big)$$

  where $\varphi = SC(\beta,\alpha,Done(accept(\beta,\alpha,p);p),q') \wedge \neg SC(\alpha,\beta,Done(acc;p),q))$, $acc = accept(\beta,\alpha,p)$ iff α offered reward q and $acc = accept(\alpha,\beta,p)$ iff β asked for reward q. This means that α revokes its commitment to give a different reward q if it asks for a reward q' from β. This ensures there only exists one offer and one reward on the table at any time.

- Asking for a reward after another proposal (with reward):

$$\bigwedge_{\alpha,\beta,q,p,p',p\neq p'} (SC(\alpha,\beta,Done(acc;p),q) \rightarrow [askreward(\alpha,\beta,p',q)]\varphi)$$

  where $\varphi = SC(\beta,\alpha,Done(accept(\beta,\alpha,p')),q) \wedge \neg SC(\alpha,\beta,Done(acc;p),q))$, $acc = accept(\beta,\alpha,p)$ iff α offered reward q and $acc = accept(\alpha,\beta,p)$ iff β asked for reward q. This says that α commits β to give reward q if it accepts the offer in p' while revoking its commitment to give q if β accepts p.

Enacting Proposals and Rewards

$$SC(\alpha,\beta,true,p) \rightarrow [p]\neg SC(\alpha,\beta,true,p)$$

where $p \in \mathcal{O}_{\{\alpha\}}$. This simply means that after the deal p is achieved the commitment is revoked. When β also fulfills its commitment to its part of the contract, we consider the contract of the agents to terminate at this point. However, in the case where a reward has been given or offered earlier, the enactment of the proposal leads to an unconditional commitment to enacting the reward as follows:

$$SC(\alpha,\beta,Done(p),q) \rightarrow [p]\neg SC(\alpha,\beta,true,p) \wedge SC(\alpha,\beta,true,q)$$

The commitment to the reward q is then revoked when the reward is enacted in the same way as for a proposal. This can be achieved simply by substituting the offer p by the appropriate reward (given or asked) that ensues from accepting a particular offer.

7.3 The Persuasive Negotiation Strategy

Given our protocol for persuasive negotiation, and knowing the effects of commitments, we now deal with the reasoning mechanism that agents must use at negotiation time to generate and evaluate offers and rewards. In particular, we do so with respect to the requirements identified by Jennings et al. (1998) (described in section 1.4):

1. Mechanisms must exist for passing proposals and their supporting arguments in a way that other agents understand — the protocol we have provided in section 7.2.3 accounts for this part of the agent's reasoning mechanism in that the agent only needs to choose the appropriate illocution to express the meaning of its offer and rewards (asked or given). The protocol also clarifies the meaning of the illocutions and their content through the commitments they entail.

2. Techniques must exist for generating proposals (counter-proposals) and for providing the supporting arguments — this demands that agents be endowed with strategies to generate offers. Here we will assume no prior information (except that of the knowledge of a conflict of preferences and the domain of discourse) about the opponent (as is commonly the case in most models (Faratin et al., 2002; Fatima et al., 2004)). In this case, the heuristic-based approach has a proven track record of eliciting good outcomes and so this is the approach adopted here (see section 1.2). Generally, these mechanisms assume no knowledge of the opponent and decide on offers and counter offers according to the behaviour of the opponent (behaviour-dependent tactics), the deadline of the agent (time-dependent tactics), and the amount of resources available (i.e. resource-dependent tactics) (Faratin et al., 1998) (see section 7.4.2.1).

3. Techniques must exist for assessing proposals (counter-proposals or critiques) and their associated supporting arguments — this means that agents need to be able to evaluate the economic benefit of proposals and rewards to them. This is normally captured by evaluating the incoming offers against the agent's preference structure or utility function. However, as we will see, in repeated encounters, agents do not know the outcome of future games a priori; that is, there exists some uncertainty about such outcomes (see section 2.3). This uncertainty needs to be taken into account in the decision making of the agents in prior games. Currently, however, there is no negotiation technique that deals with strategies specifically tailored for such repeated encounters and we aim to use persuasive negotiation to do so by minimising the uncertainty of future outcomes through the use of rewards.

4. Techniques must exist for responding to proposals (counter-proposals or critiques) and their associated supporting arguments — here again the heuristic-based models can provide good reponses to offers and counter offers. We will give special attention to those heuristic-based models that try to achieve pareto-efficiency (the Nash solution discussed in chapter 1) in the bargaining encounter such as Faratin et al.'s model (Faratin et al., 1998). In so doing, we aim to develop a bargaining mechanism that seeks the most efficient partitioning of resources.

In general, through persuasive negotiation, we give agents a means of influencing future negotiations through rewards, rather than just exchanging offers and counter offers that only impact the outcome of the present encounter. Given that negotiation normally occurs over the partitioning of some resource, the rewards, in our case, aim to constrain this partition by imposing bounds on or settling agreements on future negotiations. Thus, promises of rewards (asked for or given) partially determine the partitioning of resources to be negotiated at a later time. For example, a seller may reward a buyer with a discount of at least five pounds on her next purchase if she agrees to buy some goods at the price offered and the buyer may agree to this if she believes the discount is worth it. Similarly, a buyer might reward a seller with a guarantee to buy its next stock of goods from the same seller if a good price is offered on the current stock being negotiated and the seller may agree to this rather than continue negotiations. Such promises are important because they can result in shorter negotiations (i.e. take less time) and can lead to a more efficient partitioning of the resources (we elaborate on these in the following sections). To this end, we first develop a Reward Generation Mechanism (RWG) that generates rewards based on offers calculated by other techniques (such as heuristic-based tactics). Second, we develop a new strategy for persuasive negotiation that is specifically suited to the repeated encounters we consider. While it is possible to apply rewards to infinitely or finitely repeated games, we focus on the base case of one repetition which is simpler to analyse in order to understand the impact of rewards on the encounters. Third, we use the MMPD (as discussed in section 4.2.3) to clarify the intuition behind the generation of rewards and the selection of the type of reward to be sent. In the following sections we first build upon the definitions provided in chapter 4 to define the properties of the agents playing the iterated MMPD and offer the general intuitions behind the persuasive reasoning model. Then we describe the details of the reasoning mechanism and how agents evaluate the rewards they might receive or be asked for.

7.3.1 Properties of the Negotiation Games

We consider two agents α and β having utility functions designed as per chapter 4. In short, this means that one agent values some issues more and some issues less than its opponent. Let us assume each agent values two issues more than its opponent and two issues less (four issues to be negotiated in all). These agents are made to play two negotiation games. A negotiation game is one in which an agent (α or β) starts by making an offer over a set of issues $O \in \mathcal{O}$ and the opponent may then counter-offer or accept. The agents may then go on counter-offering until an agreement is reached or the deadline t_{dead} is reached (we superscript it with the agent identifier where needed). If an agreement is reached, the agents are committed to enacting the deal settled on according to the protocol defined in section 7.2 (if they cannot be forced to enact a deal, CREDIT can be used to check for this and alter the behaviour of the agent accordingly). We also constrain the games, and further differentiate them from the case where agents play one game each time independently of the previous one, by allowing the second game to happen *if and only if* the first game has a successful outcome (i.e. an agreement is reached within the agents' deadlines). In so doing, there is no possibility for agents to negotiate both outcomes in one negotiation round. This, we believe, more closely models realistic applications where agents will engage in a long-term relationships only if they can find some benefit in so doing given the result of their previous agreement (i.e. reach some agreements prior to continuing their relationship). Such approaches are common in long-term contracting or relationships as defined in the economic literature (Muthoo, 1999; Busch and Hortsmann, 1999).

The set of outcomes in the first game is captured by $\mathcal{O}_1$ while $\mathcal{O}_2$ represents the set of outcomes in the second game ($\mathcal{O}_n$ in the more general case). During these games, as time passes when agents exchange offers or when there is a delay until the next game, the value of the outcome decreases for each agent according to their discount factor (noted as ϵ_α for agent α).

Assuming that the time between two illocutions is τ and the time between two games is θ, the discount due to time is calculated as $\exp^{-\epsilon(\theta+t)}$ between two games and $\exp^{-\epsilon(\tau+t)}$ between offers. Note that we expect $\theta >> \tau$ generally. Obviously, the larger the value of θ or τ, the more the outcome is discounted and conversely for small values of θ or τ, since the discounting effect increases in θ and τ. The value of ϵ scales the impact of these delays, where a higher value of ϵ means a more significant discount of an offer, while a lower value means a lower discounting effect. Each agent is also assumed to have a *target utility* to achieve over the two games, noted as $L \in [0, 2]$. This target can thus be less or equal to the sum of the maximum achievable utility over the two games (2 in the case an agent has a $\epsilon = 0$ and exploits both games completely), that is $L \leq 1 + \exp^{-\epsilon(\theta+t)}$, where 1 is the maximum achievable utility in an undiscounted game (see definitions of utility functions in section 4). Finally, agents can impose bounds on the range of values for each issue they negotiate noted as $[v_{min}, v_{max}]$.

Given the above characterisation, in the next section we detail our persuasive negotiation reasoning model for agents.

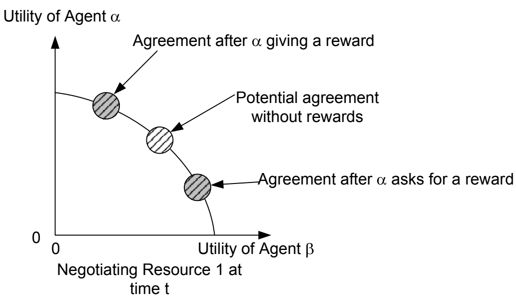
7.3.2 Applying Persuasive Negotiation

In persuasive negotiation, agents try to give rewards or ask for rewards in order to get their opponent to accept a particular offer. Rewards are about giving a higher utility outcome to an opponent in the second game (when given) or a higher utility to the agent asking for it. Agents may find an advantage to accept such rewards in the first game if it costs them more to counter-offer (due to their discount factor) or they risk passing their deadline (or their opponent's). In more detail, in negotiation a reward can be given or asked for in the following contexts:

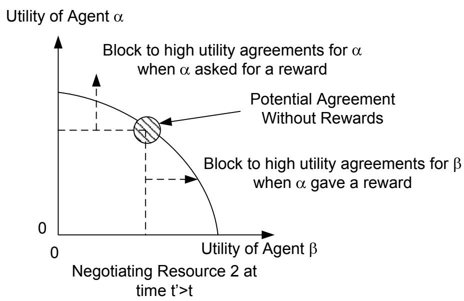
- A reward is proposed when the agent can still manage to achieve its target L after reaching an agreement *and* giving the reward. This may happen if agent α is asking β to concede in the first game, giving α more utility in the first game. Agent α may then afford to foresake some utility on the second game (which it values less due to discounting effects). It may do so by *conceding in the second game* and this acts as a reward. Note here that the reward may cost the sender something as well and it therefore needs to estimate the cost of this reward with respect to L^α properly before committing to it.

- A reward can be asked by an agent if it is able to concede in the first game so as to catch up in the second game. In this case, the agent asking for the reward has some costs in conceding in the first game and entices the opponent to pledge to something in return (a concession in the second game) for the concession in the first game. The agent asking for the reward also needs to ask for a reward that is commensurate with its target and the level of concession it is making.

These rewards do not specifically determine the outcome of the second game but specify the *negotiation ranges* that the agents will use to negotiate in the second game in a similar way to CREDIT. This is shown on figure 7.1. As can be seen in this figure,



(a) Possible outcomes of first game



(b) Influencing the second game

Figure 7.1: Determining the outcome of the second game according to the offer made in the first game.

a reward from agent α to β would be to propose a negotiation range (i.e. make offers with high utility for β) that is more favourable to β in the second game (in CREDIT, the ranges are larger for more trusted agents). The agreement reached in the first game would then be of higher utility for α. The converse applies when agent α asks β for a reward (the ranges are narrowed in CREDIT when the agent is exploited in the first game). These procedures can be seen as a trade-off mechanism often used in negotiation whereby agents trade-off gains in the present (or the future) in return for gains in the future (or in the present) (Raiffa, 1982). In general, there are two main ways agents stand to gain from using rewards as a trade-off mechanism:

1. Agents may be able to reach an agreement faster in the first game by providing some guarantees over the outcome of the second game. If some level of guarantee can be obtained for the outcome of the second game through a more favourable negotiation range, agents may find the current offer and the reward worth more than counter-offering. This, in turn, reduces negotiation time and hence the less discounted is the outcome.

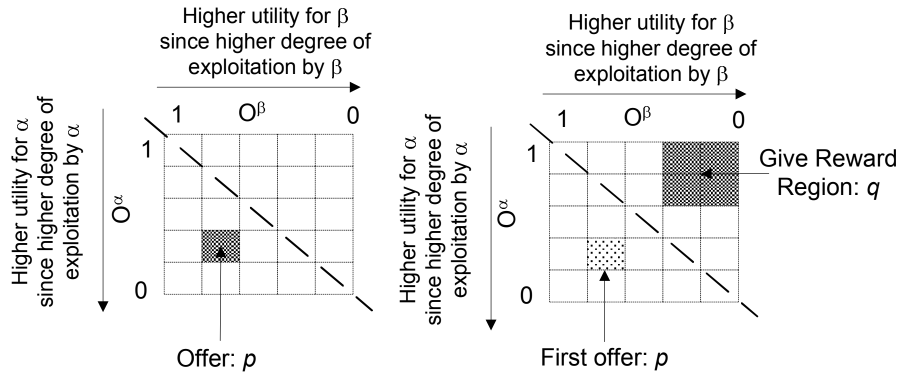
2. The negotiation mechanism can be more efficient in that the agents which value future outcomes more than their opponent are able to obtain a higher utility in future games. This may happen particularly when agents have different discount factors, such that one agent can trade-off gains in the second game, which its opponent values more, against higher profits in the first game (see discussion in section 4.2.3).

In order to allow agents to decide on what to offer or ask for as a reward, we propose that agents determine the level to which they concede in the first game in order to determine how much they will ask for or give as reward in the second game. The higher the concession, the higher will be the reward demanded, while the lower the concession, the higher will be the reward given. This is graphically illustrated in figures 7.2 and 7.3.

As can be seen in figure 7.2, α exploits β through the offer p and compensates for that in its reward q. The reward actually specifies a number of slots, one of which will be an agreement they reach after negotiation in the second game. Conversely in figure 7.3, α concedes in the first game in return for a higher utility agreement in the second game. In the next section, the exact procedure by which rewards can be calculated given the payoff structure of the MMPD.

7.3.3 Asking for or Giving a Reward

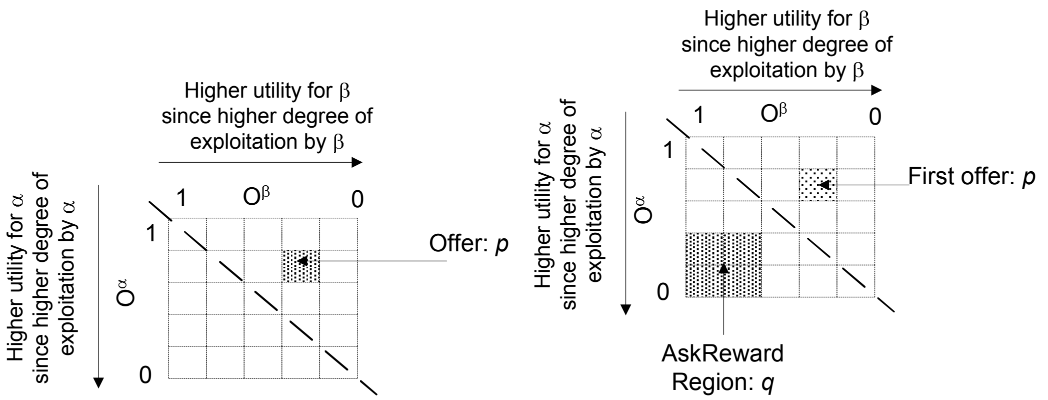
We now formalise the intuition behind the use of rewards. To this end, we extend the notation presented in chapter 4. Let $O_1 \in \mathcal{O}_1$ be an offer chosen by agent α to send to β in the first game. According to the utility functions in the MMPD, an agent values some



(a) Exploiting on first game in the offer p.

(b) Giving reward q on second game by specifying a small region over which the agreement will be made.

FIGURE 7.2: These two figures represent the value of the offer p and reward q used in $reward(\alpha, \beta, p, q)$. The offer p by α intends to exploit the opponent β and the reward q given by α to β aims to compensate for that exploitation. While 0 represents no concession, 1 represents full concession where the agent conceding gets less utility than its opponent when the latter exploits (i.e. tends to 0).



(a) Conceding on first game through offer p.

(b) Asking for reward q on second game in the shaded region specified.

FIGURE 7.3: These two figures represent the value of the offer p and the reward q used in $askreward(\alpha, \beta, p, q)$. Through the offer p, α concedes to the opponent β and the reward q asked by α aims to compensate for that concession. While 0 represents no concession, 1 represents full concession, meaning more utility for the opponent and less for the agent conceding.

issues more than its opponent and some less. Similar to the notation we use in chapter 4, $X(O^\alpha)$ denotes those issues α likes less (have a lower utility gradient than β's) in the offer $O_1 = O_1^\alpha \cup O_1^\beta$ and $X(O_1^\beta)$ denotes those issues it likes more (which β likes less). Given this, if α concedes on the values of issues in $X(O_1^\beta)$, it loses more utility than if it concedes the same amount on $X(O_1^\alpha)$. The same reasoning applies symmetrically to β. Therefore, in order to determine the level of concession in the second game on these two sets of issues, we need to transpose the level of concession from $X(O^\alpha)$ to a concession on $X(O_2^\beta)$ in the next game and from $X(O_1^\beta)$ to a concession on $X(O_2^\alpha)$ where $O_2 = O_2^\alpha \cup O_2^\beta$ and $O_2 \in \mathcal{O}_2$. In so doing, the agents will be effectively giving more or less utility to their opponent in the next game. Obviously, these decisions must be made in line with other factors such as the target L and other information or strategy the agent might employ. We will look at these in the next section. For now, we will aim to determine the level of concession which dictates whether a reward should be asked or given.

To this end, let $Con : Ag \times Ag \times \mathcal{O}_1 \rightarrow [0, 1]$ be a function that determines how much an agent concedes on an offer in the first game. The higher the value returned by Con the higher is the concession in the offer. Assuming v_{first} is the value of an issue x in O_1, then the degree of concession of α on (each) issue x in O_1 can be calculated as the relative concessions on the values taken by the issues:

$$c_x = \begin{cases} \frac{v_{max} - v_{first}}{v_{max} - v_{min}}, & \text{if } U_x(v) \text{ increasing in } v \\ \frac{v_{first} - v_{min}}{v_{max} - v_{min}}, & \text{if } U_x(v) \text{ decreasing in } v \end{cases} \tag{7.1}$$

where $[v_{min}, v_{max}]$ is the negotiation range for issue x for agent α.

Therefore, α is able to calculate how much it concedes on issues $X(O_1^\beta)$ which it prefers more than β through the following equation:

$$Con(\alpha, \beta, O^\beta) = \sum_{x \in X(O_1^\beta)} w_x c_x \tag{7.2}$$

where w_x is the weight of the issue as a level of importance of concession on it and $\sum w_x = 1$.

Now, α can also calculate how much it concedes on issues it likes less than β by replacing O^β with O^α in equation 7.2. Hence, the slot chosen in the MMPD payoff matrix is equivalent to the pair $(1 - Con(\alpha, \beta, O_1^\alpha), 1 - Con(\alpha, \beta, O_1^\beta))$ (using the axes shown in figure 7.3 for example). Therefore, the choice to give a reward or to ask a reward is carried out according to the following rule:

if $Con(\alpha, \beta, O_1^\alpha) - Con(\alpha, \beta, O_1^\beta) = 0$ **then** % equates full cooperation
  $propose(\alpha, \beta, O_1)$
else if $Con(\alpha, \beta, O_1^\alpha) - Con(\alpha, \beta, O_1^\beta) > 0$ **then** % equates to a concession
  $askreward(\alpha, \beta, O_1, aw)$
else % equates to exploitation
  $reward(\alpha, \beta, O_1, rw)$
end if

where rw is the reward given and aw is the reward asked for. In the next section we provide a number of ways of calculating these rewards. It is important to note that the agents, while knowing the structure of the pay-offs according to the shape of their utility functions (as shown in section 4.2), *do not know the exact utility* their opponent gets in the payoff matrix by virtue of their *private* discount factors and gradients of their utility functions. Thus, agents may know that they are conceding to their opponent without knowing exactly how much the concession is worth to them.

7.3.4 Determining the Value of Rewards

Having determined which type of argument *should be sent*, we can now determine the value of the reward to be given or asked. Given that an agent aims to maximise its utility in both games and, in so doing, achieve its target L, the value chosen for a reward will depend on the following factors:[3]:

- L, the target of the agent — the higher the value of L, the lower is the likely reward given and the higher the likely reward asked for (and conversely with a low value of L).
- $(Con(\alpha, \beta, O^{\alpha}), Con(\alpha, \beta, O^{\beta}))$, the degrees of concession of the agent — the higher the degrees of concession, the higher the reward asked for is likely to be and the lower the reward given (and conversely for lower degrees of concession).

We will consider each of the above points in turn.

Given an offer, an agent is able to compute the utility $l_2 \in [-, 1]$ it needs to get in the second game as:

$$l_2 = L - U(O_1) \tag{7.3}$$

where O_1 is the offer the agent has computed in the first game. Given this, we need to consider the following two cases (remember $\exp^{-\epsilon(t+\theta)}$ is the maximum that can be obtained in the second game with discounts):

- $l_2 < \exp^{-\epsilon(t+\theta)}$ — in which case it is possible for an agent to reach its target in the second game (provided the agents reach an agreement in first and it can ask for or give rewards according to the current offer or shape these to maximise its utility. As will be seen, the reward asked for or given will be constrained by the value that needs to be achieved for the target to be met in the second game (note that if $l_2 \leq 0$ the agent can afford to concede to the maximum in the second game while still achieving its target). The larger this value is, the more constrained will the second game be.
- $l_2 \geq \exp^{\epsilon(t+\theta)}$ — in which case it is not possible to give a reward but the agent might ask for a reward to achieve its target which, in this case, would be asking the opponent to concede on all issues (when $l_2 = \exp^{-\epsilon(\theta+t)}$ for example). Obviously, this may not be acceptable to the opponent if the proponent is not conceding anything in the first game. The best the agent can do is try to get an agreement on the first game to get to play the second game. In this case, the agent will simply use offers and no arguments if it is not warranted to do so.

[3]We believe these factors to be necessary rather than sufficient ones in the decision making process in the model we study.

As can be seen, the less possible it is for an agent to achieve its target in the two games, the more exploitative it may tend to be in the second game to maximise its utility. This becomes clearer when we come to determining the value of the reward according to the level of concession.

For now, assuming we know $l_2 < \exp^{-\epsilon(\theta+t)}$, it is possible to determine how much it is necessary to adjust negotiation ranges for all or some issues in the second game in order to achieve l_2. First, the agent calculates the undiscounted utility $\frac{l_2}{\exp^{\epsilon(\theta+t)}}$ it needs to get in the second game. Then it needs to decide how it is going to adjust the utility it needs on each issue, hence the value for each issue, in order to achieve the overall desired target. One approach to this is to try to gain more utility on issues the agent values more than its opponent. However, exploiting the more preferred issues can reduce the possibility of reaching an agreement (since the negotiation range would become very small and may therefore not intersect with that of the opponent). Another approach is to distribute the utility to be obtained on all issues according to the weight each issue has in the utility function (i.e. the utility to be obtained per issue is the same as the value of l_2 but is multiplied by the weight of the issue in the utility function described in equation (4.1) in chapter 4). Therefore, the required outcome v_{out} of an issue in the second game can be computed as:

$$v_{out} = U_x^{-1}\left(\frac{l_2}{\exp^{-\epsilon(\theta+t)}}\right) \tag{7.4}$$

Given the constraint defined by v_{out} (i.e. how much it should get *at least* to achieve its target), the agent needs to determine by how much it can reward or ask for a reward. To this end, the agent computes the contract $\bar{O}$ which satisfies the following properties:

$$Con(\alpha, \beta, \bar{O}_2^\alpha) = Con(\alpha, \beta, O_1^\beta) \tag{7.5}$$
$$Con(\alpha, \beta, \bar{O}_2^\beta) = Con(\alpha, \beta, O_1^\alpha) \tag{7.6}$$

This is equivalent to the procedure described in figures 7.2 and 7.3 where the level of concession or exploitation in the offer in the first game (i.e. here $O_1 = O_1^\alpha \cup O_1^\beta$) is mapped to the reward asked or given in the second game (i.e. here $\bar{O}_2 = \bar{O}_2^\alpha \cup \bar{O}_2^\beta$). Assuming linear utility functions and finite domains of values for the issues, the procedure above is equivalent to reflecting the level of concession on each issue valued more by α onto those valued more by β. This is the same as inverting equation 7.2 and finding v_{first} given c_x for each issue by inverting equation 7.1 (a procedure linear in time with respect to the number of issues considered). Let us assume that for an issue x this results in a bound v_r (a maximum or minimum according to the type of argument to be sent).

Thus, from $\bar{O}_2$, α obtains bounds on the rewards it can ask from or give to β. These bounds are expressed in terms of the minimum or maximum values (depending on which type of argument is chosen) that each issue would take in the second game. We will next consider how these bounds must be adjusted to fit the targets of the agent.

7.3.4.1 Sending a Reward

Now, given v_r and v_{out} for an issue x, assume that α prefers high values for x and β prefers low values. Also assume that it has been determined that a *reward should be sent*. Then α can determine whether a reward should be given and what is the value of the reward according to the following constraints:

- $v_r \geq v_{out}$ — α can offer a reward implying an upper bound v_r on the second game. This is because the target v_{out} is less than v_r and α can therefore negotiate with a revised upper bound of $v'_{max} = v_r$ and a lower bound of $v'_{min} = v_{out}$. The reward that can be sent in this case is the upper bound v_r implying that α will not ask for more than v_r.
- $v_{out} > v_r$ — α cannot offer a reward commensurate with the amount it asks β to concede in the first game. The best it could offer as a reward is v_{out}. This implies that α revises its negotiation ranges to $v'_{min} = v_{out}$ (with v_{max} remaining the same). In this case, the agent does not send a reward but simply *modifies its negotiation ranges*.

7.3.4.2 Asking for a Reward

Similar to the case where a reward is asked for, assuming v_r and v_{out} have been determined for an issue x, agent α has to consider the following constraints in determining whether a reward should be asked for and the value it should have:

- $v_r \geq v_{out}$ — α can ask for a reward with lower bound v_r. This is because the target v_{out} is less than v_r and α can therefore negotiate with a revised lower bound of $v'_{min} = v_r$ and the same upper bound v_{max} in order to achieve a utility that is well above its target.
- $v_{out} > v_r$ — α cannot ask for a reward with lower bound v_r. This is because it cannot achieve its target if it asks β for the exact reward it deserves. Therefore, α can privately bound its future negotiation to $v'_{min} = v_{out}$ while keeping its upper bound at v_{max}. No argument is sent in this case.

It is straightforward to extend the above reasoning mechanism to cater for β's appreciation of values of v_x (i.e. its utility is decreasing in x). Given these constraints on rewards, in the next section, we bring together the different components of the reward generation algorithm.

7.3.5 The Reward Generation Algorithm

In this section we capture the reasoning mechanism of an agent trying to give or ask for a reward. To this end, we devise an algorithm that contains all the different operations described in sections 7.3.1, 7.3.2, 7.3.3, and 7.3.4. Thus, we aim to clarify the reasoning process. We explain each significant step of the algorithm in turn (see algorithm 7.4).

As can be seen in the preconditions of the algorithm, it is required that the current time $t < t_{dead}$ such that the agent has not reached its deadline. The algorithm also requires that the agent has generated an offer O using its negotiation tactic (e.g. behaviour based, time dependent). Step 1 computes the utility that is needed in the second game l_2. Step 2 computes the minimum value v_{out} of each issue that needs to be obtained in the second game for the agent to achieve its target. Step 3 initialises values that represent outputs con_α and con_β of functions $Con(\alpha, \beta, O^\alpha)$ and $Con(\alpha, \beta, O^\beta)$ respectively. These aim to determine how much the agent concedes in the first game by virtue of the offer O. Steps 4 to 14 calculate con_α and con_β. These are calculated as the weighted sums of concessions on all individual issues (Steps 7, 9, and 11) in the offer (as in equation (7.2)). Step 13 actually maps out the concessions to the second game (as in equation (7.6)). This procedure will vary according to the particular characterisation of the negotiation game. In a zero sum game, it would simply mean inverting the level of concessions of the first game onto the second game. In contrast, in the MMPD, it equates to mapping the concessions on preferred issues of α in the first game onto those of β in the second game (as shown in equations 7.5 and 7.6). As from Step 15, the agents starts making a choice about which illocution to send and what type of argument is to be tagged to it. In case both agents are conceding an equivalent amount, Step 16 sends a proposal containing the offer O. Otherwise, it calculates the reward according to the procedure described in section 7.3.4. Steps 17 to 31 describe how an agent changes its negotiation ranges and selects a reward to be asked according to the procedure described in section 7.3.4.2. Similarly steps 32 to 46 describe the procedure for giving a reward according to section 7.3.4.1.

7.3.6 Evaluating Offers and Rewards

Having discussed how agents would generate rewards, we now describe how an agent evaluates the offers and rewards it receives. Generally, when agents negotiate through the alternating offers protocol (Rubinstein, 1982), they accept an offer only when the next offer they might put forward has a lower (discounted due to time) utility than the offer presented to them by their opponent. This is expressed as in figure 7.5.

Require: $O \in \mathcal{O}_1, \theta, \tau, \epsilon, L, R, \langle w_x \rangle, t < t_{dead}$ % O being the offer calculated according to α's negotiation tactic

$l_2 \leftarrow L - U(O)$; % utility is required in second game to meet target L

For each issue $x \in X(O)$, calculate $v_{out} \leftarrow U_x^{-1}\left(\frac{l_2}{\exp^{-\epsilon(t+\theta)}}\right)$; % values of issues required in second game

set $con_\alpha, con_\beta = 0$; % calculate $Con(\alpha, \beta, O^\alpha)$ and $Con(\alpha, \beta, O^\beta)$

for all $x \in X(O)$ **do**
    get $(x = v_{first}) \in O$; % extract values of issues in offer for first game
    get range $[v_{max}, v_{min}]$ for x; % obtain the range of negotiable values for x
    $c_x \leftarrow \frac{v_{max} - v_{first}}{v_{max} - v_{min}}$; % assuming $U_x^\alpha(v)$ is increasing in v, get concession level
    if $x \in X(O^\alpha)$ **then**
        $con_\alpha \leftarrow con_\alpha + w_x \cdot c_x$; % concessions on issues highly valued by α, $\sum w_x = 1$
    else
        $con_\beta \leftarrow con_\beta + w'_x \cdot c_x$; % concessions on issues highly valued by β, $\sum w'_x = 1$
    end if
    select O_r where $Con(\alpha, \beta, O_r^\alpha) = con_\beta$ and $Con(\alpha, \beta, O_r^\beta) = con_\alpha$; % find equivalent concession in second game
end for
if $con_\alpha - con_\beta == 0$ **then** % equal concessions from α or β
    send $propose(\alpha, \beta, O)$;
else if $con_\alpha - con_\beta > 0$ **then** % α conceding in first game - α asks reward
    for all $(x = v_r) \in O_r$ **do**
        if $v_r \geq v_{out}$ **then** % if reward falls within acceptable range
            set negotiation range in second game to $[v_r, v_{max}]$;% this is the reward asked for.
            ask-possible = true;
        else % deserved reward cannot be asked
            set negotiation range to $[v_{out}, v_{max}]$;
            ask-possible = false;
        end if
    end for
    if ask-possible **then**
        send $askreward(\alpha, \beta, O, O_r)$;
    else
        send $propose(\alpha, \beta, O)$;
    end if
else % α exploiting in first game - α gives reward
    for all $(x = v_r) \in O_r$ **do**
        if $v_r \geq v_{out}$ **then** % if reward falls within acceptable range
            set negotiation range in second game to $[v_{out}, v_r]$;
            reward-possible = true;
        else % deserved reward cannot be given
            set negotiation range in second game to $[v_{out}, v_{max}]$;% this is the reward given.
            reward-possible = false;
        end if
    end for
    if reward-possible **then**
        send $reward(\alpha, \beta, O, O_r)$;
    else
        send $propose(\alpha, \beta, O)$;
    end if
end if

FIGURE 7.4: Determining the argument type and value to be sent to β.

if $U(O_{next}) \cdot \exp^{\epsilon(\tau+t)} \leq U(O_{given})$ **then**
  $accept(\alpha, \beta, O_{given})$
end if

FIGURE 7.5: Accepting an offer in the usual case.

However, agents using persuasive negotiation also have to evaluate the incoming offer together with the reward they are being asked for or are being given. From the previous section, we can generally infer that a reward will imply a value v_r for a given issue which defines either a lower or an upper bound for that issue in the next negotiation game. For example, a reward to be given by a seller might be a guaranteed discount (i.e. a lower limit price) on the next purchase by the current buyer which could also have been a reward requested by the buyer. Therefore, given this bound, the agent may infer that the outcome of any given issue will lie in $[v'_{min}, v'_{max}]$ which might be equivalent to or different from the agent's normal negotiation ranges $[v_{min}, v_{max}]$ and may take into account the agent's target v_{out} (given its target l_2) or the value v_r itself.

Generally, we can assume that given a negotiation range $[v'_{min}, v'_{max}]$, an agent may be able to define an expected outcome of that range using a probability distribution (e.g. normal, beta) or some (fuzzy) reasoning based on its negotiation strategy (e.g. a conciliatory strategy would expect a lower utility gain in the second game as compared to a non-conciliatory one when faced with a non-conciliatory opponent). This probability distribution may be estimated from previous interactions with the agent or knowing the behaviour of its bargaining strategy and its relationship with the agent's own bargaining strategy (i.e. the relative negotiation powers of the agents as defined in CREDIT). Given this expected outcome for any issue, the agent may then calculate the expected utility (determined according to the bounds set by the reward) of that reward along with the utility of the offer to which it is tagged. Moreover, using the same procedure it can calculate the expected utility of any reward or offer that it might want to send next. By comparing the two sets of utilities, it can then make a decision as to whether to accept or counter offer in the next step. We detail such a procedure as follows.

Assume β is the agent that is the recipient of a reward (given or asked for) and that β prefers small values for the issue x being considered. Then, let β's negotiable range be $[v_{min}, v_{max}]$ for the issue x and β's target be l_2^β in the second game (which implies that it needs at least v_{out}^β for the issue in the second game).

Now, if β receives $reward(\alpha, \beta, O, O_a)$ (or $askreward(\alpha, \beta, O, O'_a)$) for the second game, O_a implies that v_r^α is the upper bound proposed for each each issue x in O_a (v_r^α would be a lower bound in O'_a). In the meantime, β has calculated another offer O_{new} with a reward O_b in which a bound v_r^β is to be given to each issue x in O_b. Then, for each issue x, β calculates the negotiable ranges given v_x^α as $[v_{min}, v_x^\alpha]$ (or $[v_x^\alpha, \min\{v_{out}, v_{max}]$ if O'_a is asked) while it calculates $[v_x^\beta, \min\{v_{out}^\beta, v_{max}\}]$ given v_x^β. We assume β can then calculate (using a probabilistic technique) the expected outcome of each range as ev_x^α

for $[v_{min}, v_x^\alpha]$ (or $[v_r^\alpha, \min\{v_{out}, v_{max}]$ in the case of O'_a) and ev_x^β for $[v_r^\beta, \min\{v_{out}^\beta, v_{max}]$. Given each of these expected outcomes for each issue, the overall expected outcomes, EO_a and EO_b, of the second game can be calculated for each type of reward respectively as:

$$U(EO_a) = \sum_{x \in X(EO_a)} w_x \cdot U(ev_x^\alpha) \tag{7.7}$$

$$U(EO_b) = \sum_{x \in X(EO_b)} w_x \cdot U(ev_x^\beta) \tag{7.8}$$

where EO_a is the expected outcome of the reward given by α, EO_b is the expected outcome of the reward given by β, $\sum w_x = 1$ and w_x is the weight given to each issue in the utility function (as per equation (4.1)).

Given that the expected outcomes have been calculated, then the agent decides to accept or counter offer using the following rule in figure 7.6 which evaluates the offer generated against the offer received to decide whether to accept the offer received or send the *reward* illocution (note the addition of discount factors to reflect the time till the next game and between illocutions, that is, sending the counter offer, receiving an accept, and sending the first offer in the second game).

if $U(O_{new}) \cdot \exp^{-\epsilon_\beta(\tau+t)} + (U(EO_b) \cdot \exp^{-\epsilon_\beta(\theta+\tau+t)} \leq U(O) \cdot \exp^{-\epsilon_\beta(2\tau+t)}) + (U(EO_a) \cdot \exp^{-\epsilon_\beta(\theta+3\tau+t)}$ **then**
  accept(β, α, O)
else
  reward$(\beta, \alpha, O_{new}, O_b)$
end if

FIGURE 7.6: Evaluating a received reward when about to give a reward

If instead, a reward O'_b were to be asked for by β along with an offer O_{new}, then β will apply a similar decision rule as above in figure 7.7 where EO'_b is the expected outcome β calculates for the reward it asks α. This rule evaluates the received offer against the newly generated offer and reward to decide whether to ask for the reward or accept.

if $(U(O'_{new}) \cdot \exp^{-\epsilon_\beta(\tau+t)} + (U(EO'_b) \cdot \exp^{-\epsilon_\beta(\theta+\tau+t)}) \leq U(O) \cdot \exp^{-\epsilon_\beta(2\tau+t)} + (U(EO_a) \cdot \exp^{-\epsilon_\beta(\theta+3\tau+t)}$ **then**
  accept(β, α, O)
else
  askreward$(\beta, \alpha, O'_{new}, O'_b)$
end if

FIGURE 7.7: Evaluating a received reward when about to ask a reward

Finally we consider the case where agent β has received a persuasive offer and can only reply with another offer without any argument. In this case, β calculates the expected outcome of the second game without any constraints (i.e. using its negotiation range

$[v_{min}, v_{max}]$ to elicit EO''_b). The rule given in figure 7.8 therefore compares the utility of the offer received against the utility of the offer generated and the outcome expected in the next game to decide whether to propose or to accept. Note here that the second game is left more uncertain in this case since the bounds have not been changed by any reward. This means that the agent cannot guarantee that it will meet its target and can also result in the agents taking more time to reach an agreement in the second game (as in the case of non-persuasive tactics as we show in the next section).

if $(U(O'_{new}) \cdot \exp^{-\epsilon_\beta(\tau+t)}) + (U(EO''_b) \cdot \exp^{-\epsilon_\beta(\theta+\tau+t)}) \leq U(O) \cdot \exp^{-\epsilon_\beta(2\tau+t)}) + (U(EO_a) \cdot \exp^{-\epsilon_\beta(\theta+3\tau+t)})$ **then**
  $accept(\beta, \alpha, O)$
else
  $propose(\beta, \alpha, O''_{new})$
end if

FIGURE 7.8: Evaluating a received reward when about to send a normal offer.

Having described our mechanisms for sending and evaluating rewards and offers, we next experimentally evaluate our model of persuasive negotiation and compare it with other standard mechanisms.

7.4 Experimental Evaluation

In this section, we describe a series of experiments that aim to evaluate the effectiveness and efficiency of our model of persuasive negotiation in repeated interactions. To this end, we build pairs of negotiating agents that respect the protocol described in section 7.2 and that use the reasoning mechanism described in section 7.3 to generate and evaluate arguments. In the following sections, we first detail the settings of the experiments and provide the results of the experiments we carry out. We also provide a new algorithm that specifically takes into account the repetitive nature of the interaction to generate rewards and offers and show how its performance compares with standard negotiation tactics that take into account one game at a time in making offers (and use our reward generation component to select rewards). Finally, we evaluate the performance of our algorithm under different conditions in order to determine the main factors that affect the effectiveness and applicability of rewards in negotiation.

7.4.1 Experimental Settings

The general settings that apply to the two negotiation games are as follows:

- The pair of negotiating agents have their utility functions shaped by the MMPD in a similar scenario to CREDIT (see section 5.4.1). The actual utility the opponent obtains for particular values of the issues are not known since utilities are private. Thus agents α and β negotiate over 4 issues $x_1, ..., x_4$ where x_1 and x_2 (e.g. price or bandwidth) are more valued by α than β while x_3 and x_4 (e.g. usage of service or time of payment), are more valued by β than α.

- the agents have their utility functions U^α and U^β specified over each issue as well as the weight of each in table 7.1.
- t_{max} — The maximum time for a negotiation game to take place is set to 2 seconds which is equivalent to around 300 illocutions to be exchanged between the two agents.[4] Unless stated otherwise, the agents' deadlines, t^α_{dead} and t^β_{dead}, are then defined according to a uniform distribution between 0 and 2 seconds.
- ϵ^α and ϵ^β — the discount factors are set to a value between 0 and 1 drawn from a uniform distribution (unless stated otherwise).
- L^α and L^β — the targets of the agents are drawn from a uniform distribution between 0 and 2 (unless stated otherwise).
- θ and $\tau - \theta$ is set to 0.5 seconds while τ is set to 0.0001 to simulate instantaneous replies (unless stated otherwise).
- $[v_{min}, v_{max}]$ — the negotiation range for each issue and each agent are defined using λ, degree of alignment of the negotiation ranges as described in section 5.4.3.2. The degree of alignment is arbitrarily set (between 0 and 1).

Agent	Utility function and weight of each issue			
	U_{x_1}, w_{x_1}	U_{x_2}, w_{x_2}	U_{x_3}, w_{x_3}	U_{x_4}, w_{x_4}
α	$0.4x_1$, 0.5	$0.9x_2$, 0.2	$1 - 0.2x_1$, 0.2	$1 - 0.6x_2$, 0.1
β	$1 - 0.2x_1$, 0.4	$1 - 0.6x_2$, 0.1	$0.9x_2$, 0.3	$0.4x_1$, 0.2

Table 7.1: Utility functions and weights of issues for each agent.

We will further assume that the first offer an agent makes in any negotiation is selected at random (but having a high utility for the agent). Also, the first agent to start any bargain is chosen at random. This random choice reduces any possible first-mover advantage a strategy may have over another (i.e. which loses less utility due to discount factors). Moreover, in order to calculate the expected outcome of the second game (as discussed in section 7.3.6), agents draw the outcome for each issue from a normal distribution with its mean centred in the middle of the agent's negotiation range for the second game with a variance equal to 0.5. Finally, in all our experiments we will use ANOVA (ANalysis Of VAriance) to test for the statistical significance of the results obtained.

[4]Preliminary experiments with the negotiation tactics suggest that if the agents do not come to an agreement within this time period, they never achieve any agreement even if the maximum negotiation time is extended.

7.4.2 Negotiation Tactics

Given that our persuasive negotiation model calculates rewards given an offer, it is possible to use standard negotiation tactics to generate the offers at negotiation time and get the corresponding reward from our algorithm (shown in figure 7.4). Here, a tactic is a mechanism that can be used (sometimes based on prior information) to generate offers or rewards in our case. To this end, we exploit the standard negotiation tactics presented in (Faratin et al., 1998).

7.4.2.1 The Standard Negotiation Tactics

We select the basic tactics that are most commonly used in the literature (Fatima et al., 2002; Faratin et al., 1998) to evaluate our model. Using such basic tactics (as opposed to more complex ones such as (Faratin et al., 2002; Winoto et al., 2004)), allows us to focus on the main properties of our PN model and to clearly show the added benefit of using rewards on different aspects of the negotiation. The basic negotiation tactics we use are described as follows:

- Behaviour based tactics (BB) — these calculate a new offer based on the difference between the opponent's last offer and its previous offer. In this way, the agent imitates its opponent. Thus, the agent calculates changes to the values from its previous offer to generate values for its next offer in the following ways for each issue:
  - Relative Tit-for-Tat (RTFT) — the change is calculated as a percentage difference between the value of the last offer and the one before.
  - Average Relative Tit-for-Tat (ARTFT) — The change is calculated as a percentage over a number of previous offers and averaged.
  - Absolute Tit-for-Tat (ATFT) — the change is calculated as the absolute difference between the value of the issue in the last offer and the one before.

  In our experiments we will calculate a BB offer by using the *average output* of RTFT, ATFT, and ARTFT.

- Time-based Tactics — these calculate a new offer by changing a previous offer depending on the amount of time elapsed since the beginning of the negotiation using a polynomial function or an exponential function. Thus, these tactics do not imitate in any way their opponent. There are two main ways these tactics operate:
  - Boulware (BW) — this only concedes significantly towards the end of the negotiation (i.e. as the agent's deadline approaches). Thus, the agent waits for the opponent to concede more until its deadline.

– Conceder (CO) — this significantly changes the previous offer early during the negotiation in an attempt to reach an agreement quickly.

As can be seen, these strategies only calculate a new offer based on the agent's own previous offer. Moreover, both behaviour-based and time-based tactics do not take into account the fact that agents are to meet more than one time and that they can either ask for or give rewards. Given this, in the next section we present a new reward-based tactic (RBT) that takes the repeated nature of negotiations into account in generating offers and rewards.

7.4.2.2 The Reward-Based Tactic

The tactics presented in the previous section usually start with an offer with very high utility for the proponent. If these tactics generate offers that are then used in our reward generation mechanism (presented in section 7.3.5), the reward generation mechanism would also start by *giving* rewards and end up asking rewards as its deadline approaches. This is because these tactics generate offers that are exploitative at the beginning of the negotiation. As the agent gradually concedes on its initial offer during the negotiation, the reward generation mechanism would ask for rewards instead. Thus, it is not possible for these tactics to ask for rewards at the beginning of the negotiation. This can significantly reduce the efficiency (in terms of the sum of utilities of the agents) of the negotiation encounter since one of the agents may be better off conceding the second game if it has a low discount factor ϵ and, in return, exploit the first game (as discussed earlier in section 4.2.3). This would mean that the more patient agent (i.e. the one with a high discount factor ϵ) could ask for a reward in the second game or the other agent could offer a reward in the second game.

Given this background, we provide an algorithm that either asks for or gives a reward at any point in the negotiation in order to reach an agreement. To do so, the agent needs to know how to evaluate incoming offers and generate counter-offers accordingly. We will consider three main cases in calculating the best response to an offer and a reward. These are:

1. An offer and a reward have been received and it is possible to counter offer with a reward.
2. An offer and a reward have been received and it is not possible to counter offer with a reward.
3. An offer has been received and it is not possible to counter offer with a reward.

We show how the algorithm deals with each of these cases in turn.

Case 1: An offer and a reward have been received and it is possible to counter offer with a reward.

In this case, an agent α needs to calculate combinations of rewards asked for or given with offers and choose the combination which it deems most appropriate to send to β. To calculate these combinations, α first needs to determine the overall utility each combination should have. To achieve this, we use a hill climbing method similar to Faratin et al.'s model (Faratin et al., 2002). In this method, the agent tries to find an offer that it believes is most favourable to its opponent while not necessarily conceding too much. In our case, this procedure equates to the agent trying to move the agreement more towards the corners (upper or lower depending on the offer and reward selected) in the MMPD. In so doing, the strategy can maximise joint gains in the repeated negotiation encounter.

Therefore, to calculate the best combination of offer and reward to send in the hill-climbing approach, the agent first calculates the utility of the next offer it intends to send and then finds the offer and reward that optimally match this utility value. By optimality, in this case, we mean that either the offer or the reward should also be the most favourable one to the opponent. Thus, the utility of the next offer is calculated according to the difference that exists between the agent's previous offer and the last one sent by its opponent and the step in utility the agent wishes to make from its previous offer. The size of this utility step can be arbitrarily set. Given a step of size f, the utility step is calculated by the function $Su : \mathcal{O}_1 \times \mathcal{O}_2 \times \mathcal{O}_1 \times \mathcal{O}_2 \times [1, \infty]$ as follows:

$$Su(O_1, O_2, O_1', O_2', f) = \frac{U(O_1)\exp^{-t} + U(EO_2)\exp^{-(\theta+t)}}{f} \\ \frac{-U(O_1')\exp^{-(\tau+t)} - U(EO_2')\exp^{-(\theta+2\tau+t)}}{f} \tag{7.9}$$

where O_1 and EO_2 are the previous offer and expected outcome in the second game from α's reward O_2 respectively, O_1' and EO_2' are the current offer and the expected outcome of β's argument O_2' respectively. When a reward is not specified by the agents, the utility calculated by the function only considers the offers made by each agent (i.e. remove $U(EO')$ and $U(EO_2')$ from its calculation).

Given the utility step Su, it is then possible to calculate the utility Nu of the combination of the next offer and reward using the following equation:

$$Nu = U(O_1)\exp^{-(2\tau+t)} + U(EO_2)\exp^{-(\theta+3\tau+t)} - Su(O_1, O_2, O_1', O_2', f) \tag{7.10}$$

Given that rewards specify bounds on the negotiation in the second game, each combination that can be offered in a step represents a space of possible agreements in the second game given an offer in the first game. Therefore, finding a combination that more closely matches the opponent's offer and reward equates to finding another space of offers that is close to the opponent's space that covers its latest offer and reward. This procedure is pictured on figure 7.9.

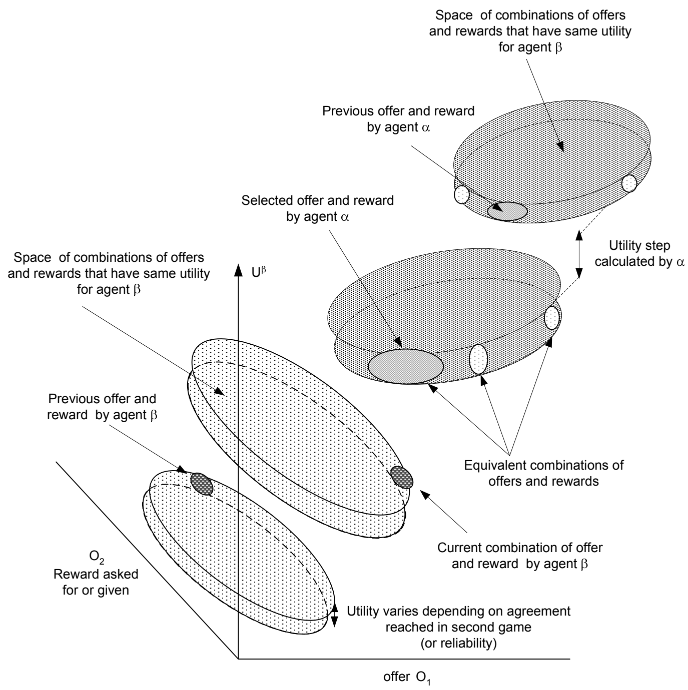


FIGURE 7.9: The hill climbing performed by RBT for an agent α to find an appropriate reward and offer in response to the offer and reward by agent β. The shaded semi circles represent spaces over which different offers and rewards have the same utility for α (top) or β (bottom). Each new offer by α is made closer to agent β's previous offer.

As can be seen in this figure, in our tactic, α calculates the most favourable combination of offer and reward for agent β that achieves the utility Nu. In so doing, our tactic aims to make offers that meet the space representing all equivalent outcomes, or the isocurve, of β in a few steps. In calculating a reward to be given we take into account the fact that in the MMPD the opponent likes some issues more than others and by maximising the opponent's gain on these issues we ensure that the reward is more attractive to the opponent. In the same way, when a reward is asked for, the associated offer is calculated such that the values of the issues in the offer are more favourable to the opponent on those issues it prefers most according to the MMPD. To calculate these offers and rewards, we use an optimisation function $OptComb : \mathcal{O}_1 \times \mathcal{O}_2 \times \mathcal{O}_1 \times \mathcal{O}_2 \rightarrow \mathcal{O}_1 \times \mathcal{O}_2$, based on linear programming (LP), that calculates the reward that is either most favourable to β or to α. *OptComb* therefore runs through our reward generation mechanism to find the best possible rewards and the associated offers whose combined utility are equal to Nu. However, *OptComb* can also *fail* to find an optimal output (as a result of the constraints being too strong (e.g. the target L being too high) or the optimizer not being able to find the solution in the specified number of steps) and in these cases, we resort to another procedure described next (i.e. Case 2).

Case 2: An offer and a reward have been received but it is not possible to counter offer with a reward.

In this case, the agent cannot find a combination of a proposal and a reward that match Nu. Therefore, the agent calculates an offer using one of the basic negotiation tactics presented in section 7.4.2.1. In this case, BB tactics would not be appropriate to generate an offer given previous offers by the opponent since these offers may also be associated to rewards. This means that the offers by themselves (which would be used in BB to calculate the next offer) do not exactly depict the concessions that the agent has made and using BB could lead to an offer where it concedes more than it should. Therefore, either BW or CO is used to generate the offer since these are independent of the previous offers made by the opponent.

Case 3: An offer and a reward have been received and it is possible to counter offer with a reward.

In the event that β only proposes an offer *without* any rewards, our tactic needs to be able to respond by a similar procedure (as in case 1) in order to continue the same step-wise search for an agreement. In this case, our tactic calculates the offer whose utility is equal to Nu (without $U(EO_2')$ in equation 7.10). Moreover, the offer calculated is such that it is the one that is most similar to the offer by β. This is achieved by running an optimization function $OptProp : \mathcal{O}_1 \times \mathcal{O}_1 \rightarrow \mathcal{O}_1$ which calculates an offer O_1 such that O_1 maximises the level of concession the opponent likes most as in the previous case while still achieving Nu. In case the issues being negotiated are qualitative in nature, the similarity based algorithm by Faratin et al. (2002) may be used.

7.4.2.3 The Algorithm for the Reward Based Tactic

In this section we describe our RBT algorithm. The algorithm is provided in figure 7.10. Here we take the point of view of an agent α trying to respond to an offer and a reward by agent β. We describe each important step of the algorithm as follows. In steps 1 to 8, α calculates the utility step by which it needs to decrement the utility of its current offer (and reward) and the value of the new offer (and reward) it needs to send, depending on whether a reward has been offered (steps 2 to 4) or not (steps 5 to 7). In step 10 α runs an optimisation to determine whether an optimal offer and reward is possible. If it is not, then in step 12, α calculates an offer using either BW or CO, while if it is, α sends the offer and reward. In the case where only an offer has been made earlier (either by α or β), the new offer is calculated using $OptProp$ in step 17 and the associated reward calculated using our reward generation mechanism presented in figure 7.4. The proposal and the associated reward (if possible) is then sent to β in step 19.

As can be seen from figure 7.10, the algorithm only generates offers and rewards in the first game. In the second game, we use a standard negotiation tactic to calculate offers. While it is possible to generate offers using the optimisation function of RBT in the second game, we do not do so in order to focus our analysis on the effect the bounds

Require: O_1, O_2, O'_1, O'_2
if O_2 and O'_2 not null **then**
  Use a probabilistic mechanism to calculate EO_2, EO'_2 (as discussed in 7.3.6) % α calculate the expected outcomes of the arguments.
  step $= \leftarrow Su(O_1, O_2, O'_1, O'_2)$ % calculate the step in utility.
  nu $= \leftarrow U(O_1)\exp^{-t} + U(EO_2)\exp^{-(\theta+2\tau+t)}$ − step % calculate the utility of the offer and reward to be generated
else
  step $= \leftarrow Su(O_1, null, O'_1, null)$ % calculate the step in utility.
  nu $\leftarrow U(O_1)\exp^{-(2\tau+t)}$ − step % calculate the utility of the offer and to be generated
end if
if O_2 and O'_2 is not null **then**
  $(O''_1, O''_2) \leftarrow OptComb(O_1, O_2, O'_1, O'_2)$ s.t. $U(O''_1)\exp^{-t} + U(EO''_2)\exp^{-(\theta+2\tau+t)} =$ nu % Here the values in the reward or the offer are optimised so as to be more favourable to the β.
  if $OptComb$ fails **then**
    use BW or CO to generate O''_1 % Resort to Standard negotiation tactics.
  else
    send offer and reward % the tactic chooses which type of illocution to use depending on whether the reward is asked from or given to β.
  end if
else
  $O''_1 \leftarrow OptProp(O_1, O'_1)$ s.t. $U(O''_1) = nu$ % find the offer that is most favourable to β given a constraint on utility.
  Find O''_2 using algorithm in figure 7.4.
  Send offer and reward if any applicable.
end if

FIGURE 7.10: The algorithm used in RBT to generate offers and rewards.

imposed by rewards have on the outcome of the second game when agents use basic tactics. We describe the experiments carried out and the results obtain in the following section.

7.4.3 Efficiency Metrics

As argued in section 1.5, PN aims to achieve *better* agreements *faster* than standard negotiation mechanisms. To test whether this is indeed the case, we first devise some metrics that help us to properly evaluate the results of our experiments as follows:

- Average number of offers — this is the average number of offers that agents need to exchange before coming to an agreement. To calculate this, we record the number of offers made each time an agreement is reached and calculate the average of these. Note that each time an offer is made a short time τ elapses. A lower average equates to a shorter time before agents come to an agreement (mutatis mutandis if the average is high). Moreover, the lower this average is the lower is the loss in utility as a result of the time-dependent discount factors ϵ. Thus we can define a *time-efficient tactic* as one that takes a relatively small number of offers to reach an agreement.

- Success rate — this is the ratio of agreements to the number of times agents meet to negotiate. The larger this success rate, the better the negotiation tactic is at finding an attractive offer for the opponent.

- Average utility per agreement — this is the sum of utility of both negotiating agents over all agreements divided by the number of agreements reached. The higher this value, the better is the strategy at finding an outcome that brings a high utility to both participating agents. Thus we define a *socially efficient* negotiation tactic as one which brings a high sum of utility in the outcome.

- Expected utility — this is equal to the average utility weighted by the probability that an agreement is reached. The probability is calculated by dividing the total number of agreements by the number of encounters agents have. Thus, if the agents find an agreement on all encounters, there is a probability of 1 that they will come to an agreement in a future encounter. A strategy with a high expected utility is one which is most likely to reach high utility agreements every time it meets other strategies.

7.4.4 Comparing PN strategies against Non-PN strategies

Having defined the tactics that agents use during negotiation and the metrics used to evaluated the process and outcomes, we can now detail the experiments carried out in order to evaluate the benefit PN brings to negotiating agents. Given our objectives set in section 1.5, we aim to show that, by using rewards which constrain the action set in the future games, agents are able to influence the outcome of negotiations and permit a better appraisal of the preferences of the agents. To this end, we experiment with the standard negotiation tactics BB, BW, CO, including those that are coupled to the RWG, as well as RBT. The settings of the strategies (i.e. the combination of tactics for the two games) for each of games played by the agents is described in table 7.2. Note that when we use the standard tactics connected to our reward generation mechanism we tag their identifiers with P (for persuasive). Thus, in the first game, NT

	Strategies		
	Non-Persuasive Agents (NT)	Persuasive Agents	
	NT	PNT	RBT
Game 1	BB, BW, CO	PBB, PBW, PCO	RBT
Game 2	BB, BW, CO	BB, BW, CO	BB

TABLE 7.2: Settings for agents' tactics and acronyms used.

(without the reward generation mechanism) is only able to make offers and evaluate offers, while PNT is able to both generate and evaluate offers and rewards. Given that persuasive strategies like PNT and RBT can constrain their rewards according to their target L (as shown in section 7.3.4), we also need to allow other non-persuasive tactics to constrain their ranges accordingly to ensure a fair comparison. Thus, we allow all tactics to constrain the ranges of the issues in the second game according to their target whenever they reach agreements without the use of any arguments (i.e. using only a *propose* illocution). The procedure to do so is similar to that described in section 7.3.4 where v_{out}, as calculated in equation (7.4), is used as the bound the negotiation range of the second game but without the use of rewards.[5]

Therefore, we postulate the following hypothesis:

Hypothesis 8. Negotiation tactics that use the reward generation mechanism are more time efficient than those that do not.

This hypothesis follows from the fact that we expect arguments to help agents find an agreement faster. To this end, we run a number of experiments with the following settings:

- We impose the following basic settings on the interactions: $L^\alpha = 0.8$ and $L^\beta = 0.8$, $t^\alpha_{dead} = t^\beta_{dead} = 1s$, $\epsilon^\alpha = \epsilon^\beta = 0.1$, $\theta = 1s$, and $\lambda = 0.8$. These settings are chosen to represent symmetric conditions for both agents and impose relatively few constraints on the two negotiation games that agents play. The symmetric nature of the interaction ensures that no tactic is in a more advantageous position to its opponent.
- We define different populations of strategies used by agents for the two games as follows:
  - Both agents use one of BB, BW, and CO in the first and second game in any combination. For example, an agent may use BB in the first game and CO in the second. We therefore play each combination of strategies against all other possible combinations. This results in 81 combinations and therefore 81 interactions per combination (one interaction being 2 games, hence an agent using one combination meets 81 other agents using similar or different combinations of strategies 162 times). We repeat these 81 interactions at least 10 times and average the results.
  - Both agents use one of PBB, PBW, and PCO in the first game and then both use one of BB, BW, and CO in the second game. This results in the same number of interactions as in the previous case. The same experimental procedure as above is repeated in this case.
  - One agent uses RBT in the first game and BB in the second game while the other agent uses one of PBB, PBW, and PCO in the first game and one of

[5]The difference between the constraint applied by the reward and by the target is that the reward applies the constraint to both agents while the constraint specified by a target only applies separately to each agent according to their individual targets.

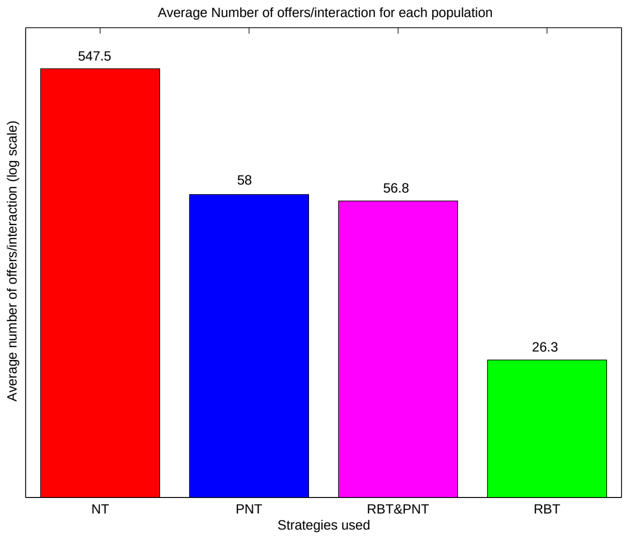


FIGURE 7.11: Average number of offers used by agents during negotiation encounters (2 games)

BB, BW, and CO in the second game. We run 81 interactions between these agents (i.e. RBT meets a P strategy 3 times) and repeat these at least 10 times and average the results. This population of strategies aims to show how effective RBT and PNT agents are at eliciting better outcomes than PNT alone.

– Both agents use RBT in the first game and BB in the second game. We run 81 interactions (each consisting of 2 games) between these two agents and average the results.

To test hypothesis 8, we ran the experiments above and recorded the average number of offers made by each population of strategies. The results are shown in figure 7.11. As can be seen, NT takes an average of 500 offers to reach an agreement, while PNT strategies take 58 and the combined PNT and RBT population takes 56 offers per agreement. The performance of only RBT strategies is significantly better than the other populations since they reach agreements within only 26 offers.[6] These results validate hypothesis 8. The superior performance of the persuasive strategies show that the reward generation mechanism helps agents to reach agreements faster. This improvement can be attributed to the fact that both negotiating agents calculate rewards and offers (through the hill-climbing algorithm) that aim to maximise their opponent's utility (as explained in section 7.4.2.2). Hence, this is faster than in the PNT and RBT case where only one party (the RBT) performs the hill-climbing properly.

The results above lead us to postulate the following hypothesis based on the fact that shorter negotiations lead to less discounted outcomes:

[6] Using ANOVA, it was found that, using a sample size of 15 for each population, and $\alpha = 0.05$, that $F = 2210 > F_{crit}$ and $p = 8 \times 10^{-74}$, hence that the results are statistically significant (i.e. the difference between the means of the distribution are not the same).

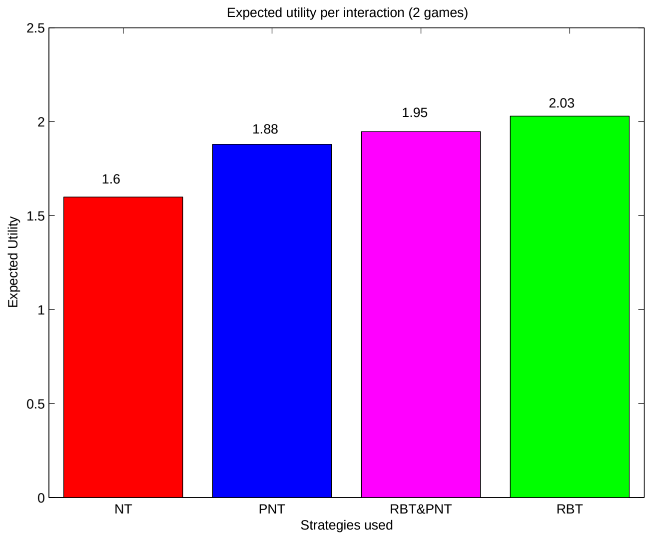
Hypothesis 9. Negotiation strategies that use our reward generation mechanism achieve a higher expected utility than those that do not.

To test this hypothesis, we run the same experiments as in the previous case and record the average utility per agreement and the number of agreements reached. Thus, it is possible to calculate the expected utility, average utility per encounter, and the success rate per game as explained earlier. The results are shown on figure 7.12.

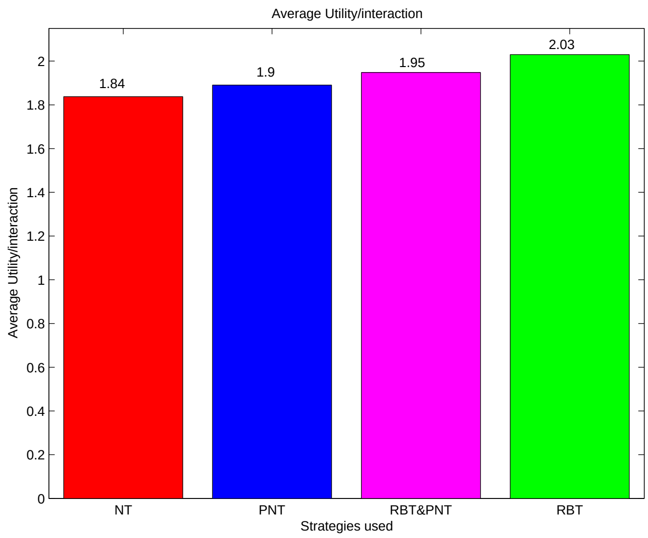
As can be seen from figure 7.12(c), the success rate of persuasive strategies is generally much higher than NT strategies (0.87/encounter for non-persuasive strategies, 0.99/encounter for PNT strategies only, 1.0/encounter for RBT and PNT, and 1.0/encounter for RBT only).[7] The expected utility shown on figure 7.12(a) followed a similar trend with NT agents obtaining 1.6/encounter, PNT 1.88/encounter, RBT and PNT 1.95/agreement, and 2.02/encounter for RBT agents only. Moreover, as can be seen from figure 7.12(b), the average utility of persuasive strategies is generally higher (i.e. 1.9/encounter for PN only, 1.95/encounter for PN and RBT, and 2.03/encounter for RBT only) than non-persuasive ones (i.e. 1.84/encounter).[8] These results suggest that PNT agents perform very similarly to NT agents when they use rewards (though rewards reduce the time to reach agreements and increase the probability of reaching an agreement). As discussed earlier in this section, PNT agents usually generate offers first (starting from high utility ones as for NT agents) and then calculate the rewards accordingly. Given this, the agents tend to start by giving rewards and end up asking for rewards. As the negotiation proceeds (if the offers are not accepted), the offers generally converge to a point in the middle of the MMPD and rewards converge to a *region* around the centre of the MMPD. This process results in a lower overall utility over the two games than if each agent exploits the other on each game in turn. However, when PNT agents use rewards they are able to reach agreements much earlier on during the negotiation so that the outcome results a more efficient partitioning of the resources and hence a higher utility than NT agents. If rewards are selected in a more intelligent fashion, as in RBT, the agents reach much higher overall utility in general. This is further demonstrated by the results of the RBT agents which efficiently select offers and rewards and therefore tend to reach agreements that have high utility for both participating agents. Given this, we can infer that the reward generation mechanism used together with normal strategies which do not fully exploit the potential of rewards in reaching agreements allows agents to reach better agreements to some extent and these agreements are reached much faster and more often!

[7] Using ANOVA, it was found that for a sample size of 15 for each population of PNT, PNT and RBT, and PNT only, with $\alpha = 0.05$, $F = 8.8 > F_{crit} = 3.15$ and $p = 4.41 \times 10^{-4}$. These results prove that there is a significant difference between the means of PNT and the other strategies. The success rate of NT agents were always lower than the other populations in all elements of the sample.

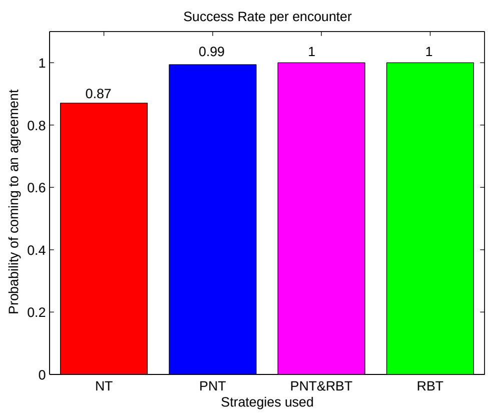
[8] These results were validated statistically using ANOVA, where it was found that $F = 3971 > F_{crit} = 2.73$, and $p = 7.36 \times 10^{-80}$, for a sample size of 15 per population and $\alpha = 0.05$. These results imply that there is a significant difference between the means of the populations.



(a) Expected utility obtained by two agents per encounter.



(b) Average utility obtained by two agents per encounter.



(c) Success rate of strategy per encounter.

FIGURE 7.12: Expected utility, Average utility and success rate of agents using different strategies.

It can also be noticed that the performance of mixed populations of RBT and PN agents performs less well than RBT agents and slightly better than a pure PN population (see results above). This suggests that the RBT agents can find agreements that convince their PNT opponent more quickly as they are able to propose better rewards and offers than PNT agents. Moreover, it was noted that on average, both RBT and PN agents obtained equal average utilities per agreement (i.e. 0.96/agreement)[9] This also suggests that RBT agents can avoid exploitation by any other PN-based agent. This is because the hill-climbing mechanism of RBT agents calculates offers that can convince the opponent without reducing the utility of RBT (and PNT) agents significantly (i.e. in small steps).

In general, through the above experiments we have empirically proven the usefulness of rewards in bargaining. Thus, we have achieved our initial aim of using PN to enable agents to achieve better agreements faster. In the following section, we further study our RBT strategies to see how it is affected by different conditions in the environment.

7.4.5 Evaluating the Reward Based Tactic

In this section we further explore the properties of RBT by studying its behaviour when key attributes of the agents are varied. As can be deduced from section 7.3, there are a large number of attributes that can affect the behaviour of RBT but here we will focus on the following main ones which we believe have a siginificant impact on both our reward generation component and the behaviour of RBT. These attributes are:

1. L — the target determines the size of the reward that can be given to or asked for as determined by v_{out} in equation (7.4) and the procedure described in equation 7.3.4. Given this, varying L allows us to study the effectiveness of PN in general as the possibility of asking for or giving a reward changes. Moreover, we aim to study the effect of one agent having a lower or higher target than its opponent on the outcomes of negotiations.

2. ϵ — the discount factor dictates the utility of offers as well as rewards. In particular, we aim to see how RBT and our reward generation mechanism can help agents that have different discount factors find good agreements.

3. θ — the delay before the second game is played determines the value of the reward. Increasing this value can significantly reduce the value of a reward to an agent. By varying θ we aim to see how it impacts on the use of rewards during negotiation and how this affects the outcome of each game.

[9]We validated this result using ANOVA with a sample of size 15 per strategy and $\alpha = 0.05$. Thus it was found that the null hypothesis (i.e. equal means for the two samples) was validated with $F0.13 < F_{crit} = 4.10$ and $p = 0.71 > 0.05$.

First we investigate the impact of the negotiation target L on the outcome of negotiations. In this context, L is used to decide whether a reward should be sent or not and what the negotiation ranges of an agent should be in the second game (see section 7.3.4). The higher the value of L, the less agents are likely to be able to construct arguments. This is because, an agent may have to shrink the negotiation range in the second game more in to achieve a higher L over the two games. Therefore, we expect the agents to achieve fewer deals and have a corresponding lower overall expected utility. Moreover, in the case where only one agent has a high L, then the opponent's rewards are less likely to be accepted because these rewards are less likely to allow the agent to achieve its target, and hence the agents are less likely to come to agreements or take more offers to come to any agreement. In this case we would also expect the agent with the higher L to negotiate more strongly and constrain the second game more such that it should get a higher utility than its opponent. To investigate these intuitions, we will consider a pair of agents α and β that use RBT and postulate the following experimental hypothesis:

Hypothesis 10. The higher the value of L^α relative to L^β, the higher is the average utility α compared to that of β.

To test hypothesis 10 we ran an experiment where the agents were made to negotiate using similar settings as in the previous section except to the fact that the target of α was made to vary between 0 and 1.5 while β's target was kept fixed at 0.5. The results of the experiment are shown in figures 7.13,7.14, and 7.15.

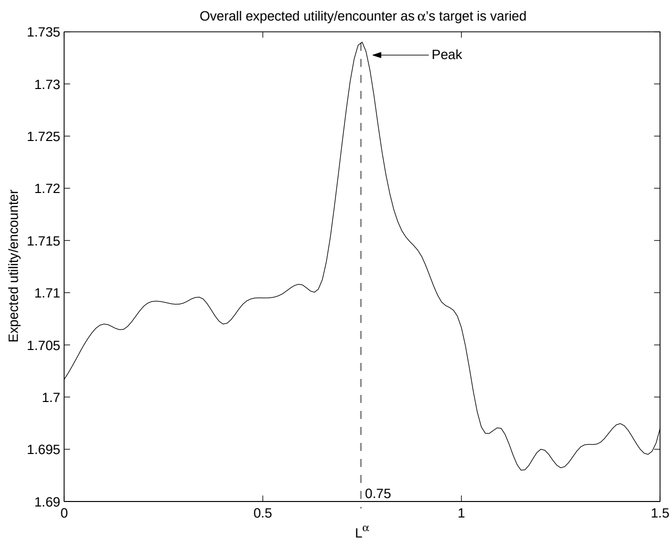


FIGURE 7.13: Expected utility of α and β when $L^\beta = 0.5$ and L^α are varied.

As can be seen from figure 7.13, the overall expected utility of both agents suffers a sharp drop after a peak at $L^\alpha = 0.75$ and there is a a sharp rise in the number of offers exchanged between the two agents (in figure 7.14). Moreover it was found that the success rate of the agents did not significantly drop (decreasing from 1 when $L^\alpha = 0.75$ to around 0.99 till $L^\alpha = 1.5$). The main cause for the drop (and peak) in expected

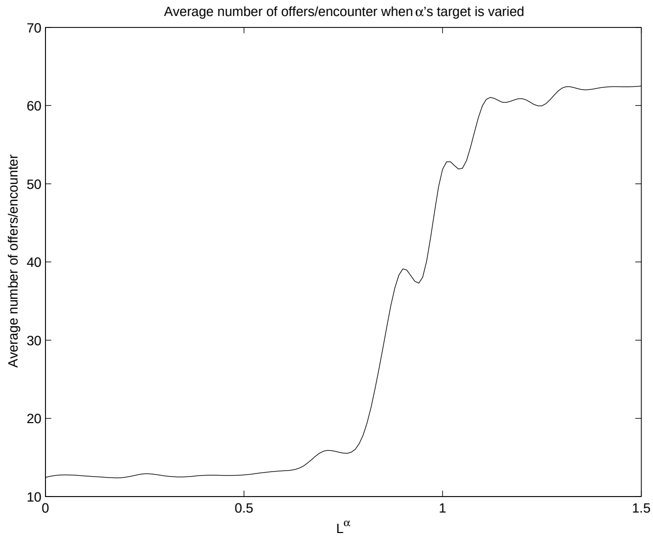


FIGURE 7.14: Average number of offers between α and β when L^α is varied.

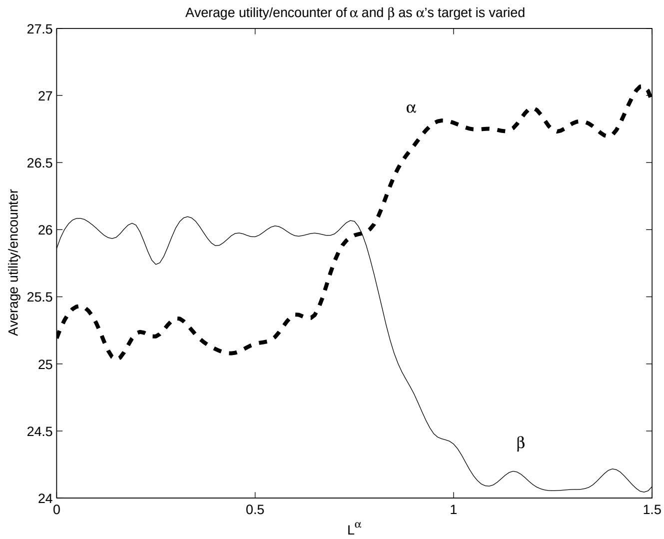


FIGURE 7.15: Average Utility of α and β when $L^\beta = 0.5$ and L^α is varied.

utility and rise in the average number of offers can be explained by the results shown in figure 7.15. As can be seen, as from $L^\alpha = 0.75$, α's utility gradually rises while β's utility sharply falls. This means that α exploits β on all the issues that are negotiated.

In more detail, in order to obtain $L^\alpha = 0.75$, α would need to exploit β in the first game on all the issues it prefers more than β or exploit β on all issues (which it likes less or more than β) in the second game. This can be deduced from the weights used in the utility functions shown in table 7.1. Therefore, at this point, α and β are likely to exploit each other maximally on the issues they prefer in each game. This results in a high point in utility since it represents the cooperate-cooperate point in the MMPD (hence the peak in figure 7.13). When $L^\alpha < 0.75$, the agents can still find agreements without completely exploiting their opponent on any issue and therefore agree to proposals and rewards that result in a lower overall utility since the outcome then lies further away from the cooperate-cooperate point of the MMPD.

Beyond $L^\alpha = 0.75$, it becomes harder for α to give or ask for any rewards. This is because, as L^α increases, the use of arguments decreases as the α's ability to concede in either game decreases (since it nees to achieve a high target) and α can only constrain its negotiation ranges more and more in the second game in trying to get its target. However, given that β has a low target, it can still afford to be exploited by α and still manage to reach its target over the two games. Hence the success rate of the two agents does not significantly decrease. However, given the more stringent demands of α, the agents are likely to exchange a large number of offers (i.e. β conceding a significant number of times) until an agreement is reached.

In general, these results validate hypothesis 10 and also confirm our intuition that α's bargaining power should increase with respect to its target. Given these results, it can be expected that if the second game were less discounted, α could have started exloiting β at a higher value than 0.75. We will therefore explore such discounting effects on the negotiation and investigate the effect of increasing both agents' targets at the same time to see the general behaviour of the system as the discounts and targets are varied.

Before doing so, however, we next study the effect of the discount factor ϵ^α on the outcome of the negotiation (keeping $\epsilon^\beta = 0.5$). In this case, a low value of ϵ^α equates to a low discounting effect on the outcome of the two games and conversely for a high value of ϵ^α. Therefore we can expect that as ϵ gets higher the agreements reached in the two games would be much more discounted and hence result in a lower overall expected utility. Moreover, with higher ϵ values, agents will find it harder to achieve their target L as they will value both offers (and counter offers) and rewards less. Agents are then likely to take more offers to reach an agreement and reach fewer agreements as well. In the case where only α's discount factor ϵ is varied, we would expect that the agent with the higher discount factor would be more likely to accept any offer by its opponent since counter-offering might take up time that discounts its own offer more than the one offered by the opponent. This means that the more patient agent is likely to get its offers more easily accepted (i.e. take fewer numbers of offers on average) and exploit its opponent more. Hence, as predicted by game theoretic models of bargaining (Muthoo, 1999), the more patient agent gets an increasingly higher average utility than its less patient opponent as the difference between their discount factors increases. We therefore postulate the following hypothesis:

Hypothesis 11. The higher the value of ϵ^α relative to ϵ^β, the less agents are likely to reach agreements and take more offers to reach an agreement.

To test this hypothesis, we ran a similar experiment as above apart from the fact that we kept the target for both agents at $L^\alpha = L^\beta = 0.5$ and we varied ϵ^α between 0 and 3 (while keeping $\epsilon^\beta = 0.5$). In this context, it is obvious that the overall expected utility of the agents will decrease when ϵ^{α} increases (and the utility α gets decreases as a result of the discounting effect). Given this we recorded the average utility of each agent and the number of offers they take to reach an agreement. The results are shown in figures 7.16 and 7.17. As can be seen from figure 7.16, β's utility gradually decreases as ϵ^{α} rises.

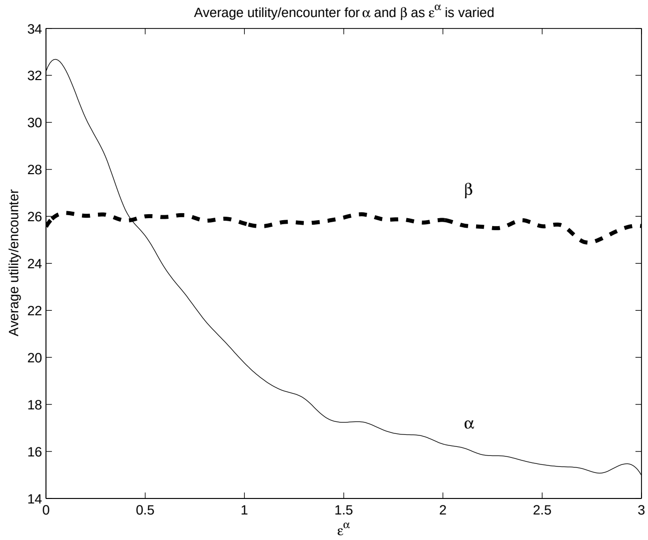


FIGURE 7.16: Average utility of α and β when ϵ^{α} is varied.

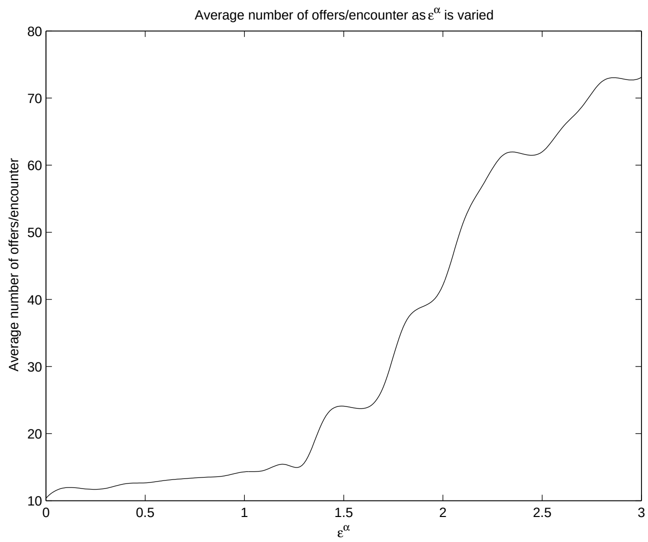


FIGURE 7.17: Average number of offers made by α and β as ϵ^{α} is varied.

The number of offers used by the agents also rises significantly as ϵ^{α} increases beyond 1.44. This is because, beyond $\epsilon^{\alpha} = 1.44$, the discounting of the second game is such that it is worth less than 0.5 (assuming α exploits all issues in the second game). Thus, it becomes impossible for α to ask for rewards and it can only rely on giving rewards. Moreover, as the discounting effect increases, it also becomes harder for β to convince α with rewards. Eventually, as time passes, the agents can only rely on simple proposals and α constrains its negotiation ranges in the next game so as to achieve its target.

Given this, negotiations take even more time in the second game (as in the previous experiement). Therfore, the target reduces the advantage of β's patience (i.e. in having a lower discount factor) in this type of game. It was also found that the success rate of the agents does not significantly decrease (from 1 to 0.98) after $\epsilon^\alpha = 1.44$. This suggests that β concedes more than α in the second game in order to come to an agreement. This is also confirmed by the β's decreasing average utility in figure 7.16. These results therefore validate hypothesis 11.

Given the above results, we can expect that the combined effect of an increasing target and an increasing discount factor should significantly reduce the expected utility of both agents and increase the number of offers they need to make to come to an agreement. We therefore postulated the following hypotheses:

Hypothesis 12. The higher the value of L^α and L^β, the lower expected utility of both agents.

Hypothesis 13. The higher the value of ϵ^α and ϵ^β, the less agents are likely to reach agreements and take more offers to reach an agreement.

Therefore, we varied both agents' discount factors and targets to see which had a stronger effect on the negotiation outcomes. The plot of the expected utility of the agents is shown in figure 7.18.

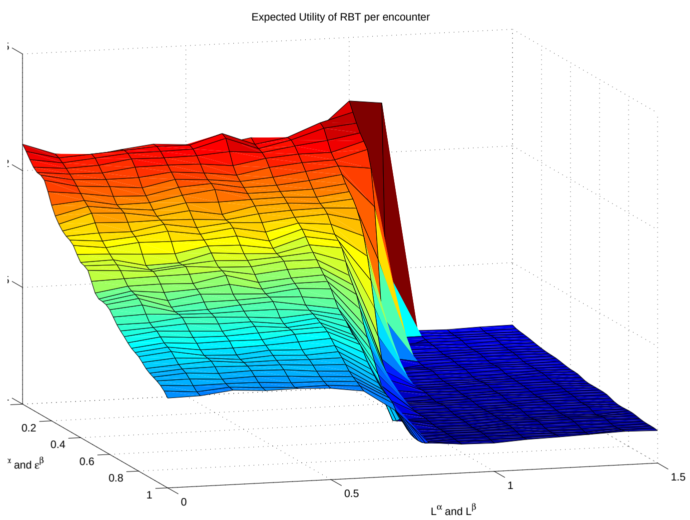


FIGURE 7.18: Varying the target and discount factor of α and β and the resulting expected utility.

As can be seen from figure 7.18, the expected utility is more significantly affected by the target of α and β. The results confirm hypotheses 12 and 13. Indeed, the drop in utility (as in the experiment for hypothesis 10) is noticed at particular values in the agents' target, corresponding to points where the target can no longer be met easily as a result of the second game not providing sufficient utility. Moreover, we notice that the point at which expected utility drops relative to target values decreases in ϵ. This confirms our initial intuition that the discount factor influences to some extent the effect of the target on the expected utility.

We also recorded the average number of offers made by the agents to see the impact of the target and discount factors on it. The results are shown in figure 7.19. As can be seen from figure 7.19, the drop in expected utility is reflected by the jump in the number of offers made. The region where the peak occurs corresponds to values of the targets and discount factors where the agents are still able to use rewards to persuade each other and significantly shrink their negotiation ranges in the second game to reach their target. Beyond this peak (i.e. for higher values of the targets in particular), the agents can only find agreements in the first game and they do so according to the hill climbing mechanism of RBT (which guarantees that they meet in a few number of steps). Note that the plateau at low values of L is at a lower value than that at high values of L, suggesting that rewards can significantly reduce the number of offers made to reach an agreement compared to those that only make offers using the hill climbing method.

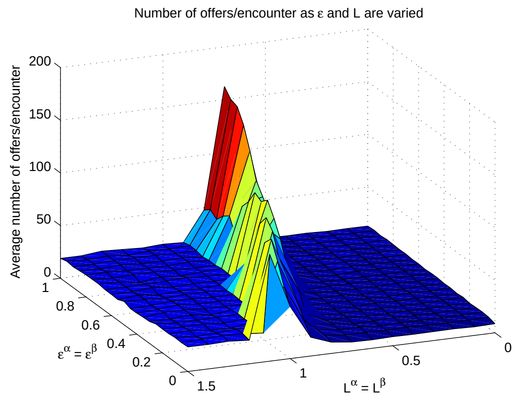


FIGURE 7.19: Average number of offers when L and ϵ are varied.

Finally, given that higher values of ϵ on the offers and rewards decrease the probability that agents reach an agreement and increase the number of offers exchanged, we expect a similar effect for higher values of the delay. This is because a longer delay decreases the value of rewards to both agents, and hence reduces the probability of reaching each agent's target L. Therefore, we expect that the longer the delay θ, the lower the success rate of the agents and the higher the average number of offers needed to reach an agreement. Given this, we postulate the following hypothesis:

Hypothesis 14. The higher the value of θ, the less likely it is that agents will reach profitable agreements and the more offers they take to reach an agreement.

As for the above hypotheses, we ran a similar experiment keeping $L^\alpha = L^\beta = 0.5$ and $\epsilon^\alpha = \epsilon^\beta = 0.5$, varied θ between 0 and 10 seconds, and recorded the expected utility of the agents. The success rate of the agents only decreased slightly from 1 to around 0.99 after $\theta > 5$ while the number of offers significantly increased when θ increased beyond

3 seconds as shown on figure 7.20. These results confirm hypothesis 14. The reason

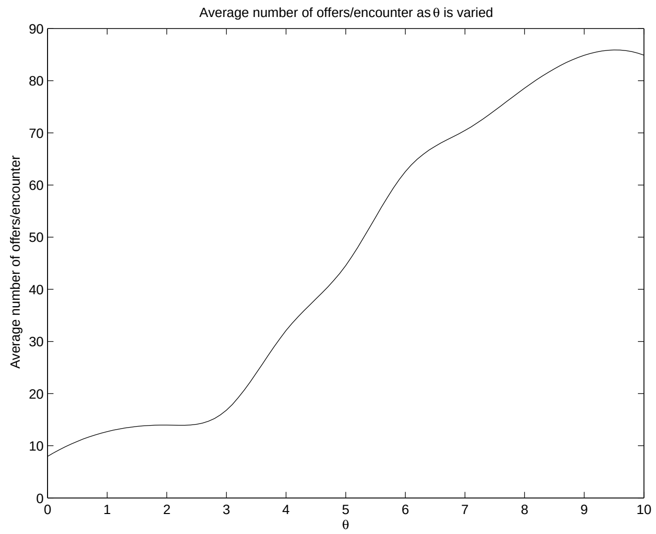


FIGURE 7.20: Average number of offers per encounter as θ is increased.

for the jump in the number of offers at $\theta = 3$ has a similar explanation to that in the previous experiment for $\epsilon^{\alpha} = 1.44$. Indeed at $\theta = 3$, the total value of the second game decreases below 0.5 and decreases the value of rewards that can be given or asked for. This results in the agents only being able to make offers without arguments and hence increase the constraints on the second negotiation and increases the number of offers needed to reach an agreement. To confirm these results, we also recorded the number of agreements reached through the use of rewards. As shown in figure 7.21, it was indeed found that the number of agreements reached through the use of rewards decreases as θ increases.

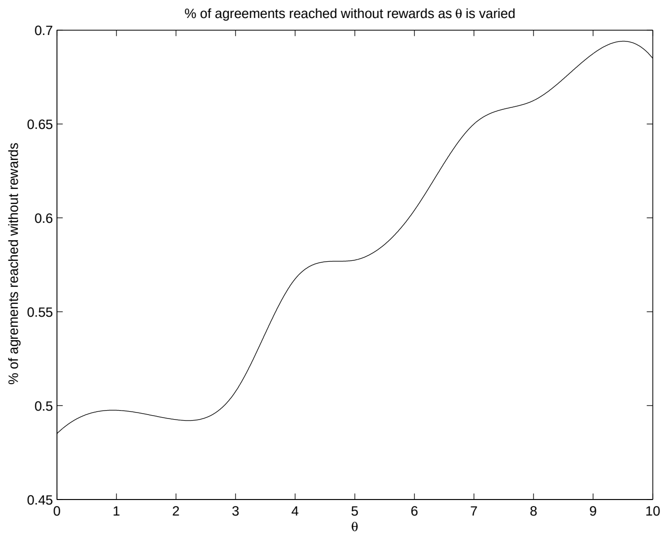


FIGURE 7.21: Percentage of offers made without rewards as θ is varied.

7.5 Summary

In this chapter we have presented a comprehensive model of persuasive negotiation. In particular, we have provided a novel protocol based on dynamic logic to specify commitments that arise in persuasive negotiation based on the exchange of arguments in the form of rewards (given or asked for). Given this, we specified a new decision making model for agents to generate, select, and evaluate rewards that they exchange. This mechanism allows agents to reduce the uncertainties about the action set and the preferences of the agents (as per our objectives set in chapter 1). Thus, it is shown that the use of rewards can result in agreements with higher expected utility than standard negotiation tactics and that it can take less time in doing so. Moreover, we developed a new negotiation tactic specially suited for persuasive negotiation and showed how it can allow agents to reach better agreements than standard negotiation tactics augmented with our reward generation mechanism.

Given this, we can broadly conclude that PN effectively enhances the search for agreements in negotiation. Moreover, we have shown how PN can be efficiently applied to influence outcomes in repeated encounters. Given this, in the next chapter we show how PN can be practically applied to allow agents to negotiate over repeated encounters in a flexible manner through the use of rewards. Moreover, we aim to show, through this application, how CREDIT can also be used alongside PN in such encounters in order to reduce the uncertainty agents have about the honesty and reliability of their counterparts.

Chapter 8

Persuasive Negotiation in a Pervasive Computing Environment

Having developed CREDIT and our model of PN, we aim to show how these can be integrated in a practical application as per our initial objectives stated in chapter 1. In particular, we apply PN to the problem of managing the display of notifications in a pervasive computing environment. In this context, agents have different (private) preferences (built in by their human users) about the notifications that are received at different points in time. Thus, while some messages might be preferred by the user receiving the notification, it might be intrusive to the group activity the user is engaged in, hence less preferred to the other group members. In other cases, however, the message might be preferred by the group as well. Given this, agents are faced with significant degrees of uncertainty about the preferences of their opponents (since these are kept private) and notifications that may arrive in at any given moment during the group activity. Hence, the distributed and repeated nature of these interactions make it suitable to apply CREDIT and PN to reduce these uncertainties (as per the attributes of CREDIT and PN discussed in section 1.5). Using these models, we therefore show how they can help selfish agents to manage the intrusiveness of notifications on their human owners' group activities.

The rest of the chapter is structured in the following way. Section 8.2 describes the notion of interruptions and defines intrusiveness for pervasive environments. It also describes the context of the meeting room scenario that we use to demonstrate our solution. Section 8.3 provides an account of our agent-based solution, while section 8.4 describes a practical implementation of our solution. Finally, section 8.5 summarises the main achievements of this chapter.

8.1 Introduction

Pervasive computing artefacts such as laptops, smart whiteboards, video phones, and pagers are becoming increasingly commonplace in our every day lives (Abowd and Mynatt, 2000; Chandy et al., 2002). Moreover, such devices are becoming increasingly interconnected given advances in communication technology (e.g. 3G mobile phones, bluetooth) and processing power (e.g. PDAs, Video telephony). Thus, users of such devices can be contacted in very many ways and in most environments.

There are a number of advantages to this. First, users are able to receive information on a variety of interactive media which afford different types of interactions (e.g. responding to an email, responding to a video call). Second, users can communicate information through many different light and portable devices that can be used anywhere with such connectivity (e.g. GPRS palm, wireless laptop). Third, users can use these devices as supports to their tasks (e.g. a stock trader using different monitors to check stock prices, while at the same time having a phone call with a broker, or a customer checking prices of books online on a PDA while walking in a bookshop to check which books are better deals). Thus, in general, such technology can increase the efficiency and well-being of its users.

However, the uses of such pervasive technology also have some downsides. First, notifications or messages received on such devices disturb the users in their current focus of activity which might warrant more attention than the message itself (e.g. a phone call received while making a presentation or an instant messenger (IM) beeping while having a discussion). Second, this shift of focus affects the other users with whom the user is interacting (e.g. the attendees of the presentation lose track of what is being presented or the discussion stops). Third, using current filtering techniques (e.g. in instant messengers or phones) it is not possible to distinguish between messages which are completely irrelevant to either the current activity of the users (e.g spam mail, wrong number phone call), as captured by their *context*, or their own interests (e.g. a subscribed weekly electronic newsletter, or news flash), and messages which are actually relevant to the preferences of the users and/or help in the task at hand (e.g. an email containing attachments that need to be used in a presentation, or a phone call from the users' boss).

Given this background, there is a clear demand for middleware systems to manage the *intrusive* nature of interactions in pervasive computing environments. Such systems should nevertheless permit users to carry out their normal activities and effectively interact with pervasive computing artefacts seamlessly without blocking incoming information that might be important given the interests of the user and their context. In more detail, these systems need to be dynamically configurable so as to adapt to the current context of the user and their interests. For example, the underlying system should be able to react differently when the user is in a meeting (where notifications should be relevant to the meeting or be important for the user) and where the user is alone and browsing an email (when emails that are not very important can be viewed). Moreover, in order for the system to be non-intrusive itself, it should be able to autonomously decide on behalf of the user which is the best course of action, given the objectives of the user and other users that may be in the same environment.

Given these desiderata, agent-based computing has been advocated as the natural computation model for such systems (see chapter 1). More specifically, pervasive computing environments can be modelled as open MAS that are composed of autonomous software agents that each represent their respective human owner and make decisions on their behalf given their specified preferences. Thus, in our model, users relinquish the management of incoming messages to their *software agent* which decides when, how, and where messages are to be displayed such that the notification delivered disturbs the user on the right device, given the intrusive nature of the device and the level of intrusiveness permitted by the user's context (e.g. an unimportant instant messenger chat window may be hidden until the meeting is over, while an important email might be highlighted in the list of received emails with a beep to warn the user). As part of this endeavour, the agent may need to negotiate the display of notifications (i.e. on which device and at what point in time) with other software agents that represent other users in the environment in order to reconcile the preferences of the group, as opposed to those of the user, when the latter is involved in a group activity.

In more detail, agents need to negotiate about each notification received since they are uncertain about the preferences of other agents about the notifications received (since the other agents' preferences about the contents are kept private) and about new notifications they may receive as the meeting progresses (which may disturb the meeting more if they occur too frequently). For example, if a video call expected by the group is received on one user's laptop, other agents may negotiate to have it displayed on the public display which can be viewed by all participants in the meeting. Conversely, if no one is interested in an instant messenger message received by a participant, then the other agents would negotiate to make sure that message could be redirected to his email if it is not important or beeped to him if it is. Thus, through negotiations, agents can discover that they have similar preferences about notifications received or can even persuade each other through the use of arguments (see chapters 2 and 7) to accept notifications. For example, one agent can accept a particular notification if the proponent of a notification agrees to accept some future notifications by the former in future. In so doing, the agents can give rewards to each other by agreeing to the display of future notifications in exchange for displaying a particular notification in the present encounter. Also, agents can also ask for rewards by asking opponents to agree to display of future notifications in exchange for agreeing to their notifications in the present encounter. Thus, agents can reduce the time they take to reach an agreement the next time a notification is received since uncertainty about the space of offers is reduced in the next encounter.

Against this background, this work advances the state of the art in the following ways. First, we define a typology of interruptions for pervasive computing environments using notions of intrusiveness. Second, using our model of PN and CREDIT we develop a novel agent-based solution to the problem of managing intrusiveness given the preferences of the human users. Finally, we describe an implementation of our system in a meeting room scenario using the Jabber platform as the underlying architecture of our solution.

8.2 Intrusiveness and Interruptions

McFarlane was the first to distinguish the notion of intrusiveness from that of interruption (McFarlane, 1999). He defines the former as *the degree of interference with the realisation of the main task of a group caused by a number of intrusions.* In turn, an intrusion is defined as *an occurrence of a process or event that is not intimately related to the current task of a group and that interferes with the realisation of that task.* Note that *interruptions* and *intrusions* are clearly distinct concepts: the latter cause errors where people incorrectly perform actions in an interrupted task after task switching (i.e. handling the interruption), while the former are general methods by which a person shifts his focus of consciousness from one processing stream to another (McFarlane, 1997). Thus intrusions can be regarded as a subset of interruptions (see section 8.2.2 for more details).

8.2.1 Receiving and Managing Interruptions

Interruptions can happen in very many ways. Specifically, in pervasive computing environments, these interruptions generally take the form of notifications that are received on the various artefacts that a user may possess or perceive in his environment. To this end, McFarlane identifies four main ways to disrupt someone (McFarlane, 1999) and we identify examples where these apply in pervasive computing environments:

1. *Immediate*: require the attention of the user immediately without any other choice. This might involve displaying a notification on a public display or popping up a chat message in an instant messenger when a message is received.

2. *Negotiated*: allow the user to choose the moment when they will deal with the interrupting activity that needs attention. A user may thus notice that an email has arrived on his email client or that a message is flashing on his instant messenger.

3. *Mediated*: alert the user on another device rather than the one on which it was supposed to be delivered. Such systems are now starting to become reasonably

standard. For example, an email client can redirect via SMS (Short Message Service) to a phone or a phone call is re-routed to the voice mail of the user (which he can access at a later time or listen to after the message is recorded).

4. *Scheduled*: come at prearranged intervals. For example, a user may have a pre-arranged video-conference call or may schedule a periodic alarm on a PDA to alert him to take his regular insuline dose.

Whatever the form in which a message is received, there are four possible responses to it (Clark, 1996):

1. *take-up with full compliance* — handle the interruption immediately.
2. *take up with alteration* – acknowledge the interruption and agree to handle it later.
3. *decline* — explicitly refuse to handle the interruption.
4. *withdraw* — implicitly refuse to handle the interruption by ignoring it.

In each of the above responses, some degree of mental processing by the user is involved in deciding what course of action to take. In most cases the answer depends on the preferences of the user with respect to the information available about the content of the notification. Typically, the information available from the notification (rather than from the whole content of the message) is the name or identification number of the message sender and a subject line briefly describing the content of the message (mostly in emails, IM messages, and sometimes on video conference calls as well). From this information, the user can usually tell whether the message (for which the notification has happened) is something that he asked for (e.g. information about his children's health), or was sent to him to inform him of something important (e.g. an email about his latest stock prices), or is relevant to his current context (e.g. an advertising SMS received on his mobile phone about a shop in his surroundings). The device through which the notification is conveyed determines the degree to which the user is disturbed (e.g. a notification displayed on a public device on which a message is publicly visible is almost certain to alert the user and other users present, while a message shown in an email client without beeping or popping up an icon, is sure not to disturb the user). Moreover, the device gives a level of guarantee that the notification will be seen by the user (e.g. an IM beeping is sure to alert the user, while the user must be looking at his laptop screen to see the heading of an email). Thus, the right device must be chosen in the right context in order *to balance the importance of the message with the intrusive nature of the interaction with these devices* (e.g. an IM must not beep when the user is doing a presentation, while he can be beeped when he is not focussed on any important task). In the next subsection we therefore consider the issues involved in choosing devices that can be used to disseminate the information contained in the message.

8.2.2 Typology of Interruptions

We can generally assume that interruptions define a means of disseminating information[1]. Now, whether this information warrants the disturbance of the user is dependent on the relevance of the information to the needs and preferences of the user or the user's group. We therefore classify the information dissemination solutions as information *push*, where information is not expected by the user, or *pull*, where the information is expected.

Whenever messages are received, we will use the preferences of the users to define the messages' pull or push nature. Thus, whenever preferences specify that a sender and a particular subject is much liked, then the message concerned is considered to be pulled, while if preferences do not specify the sender or the subject then that message is considered to be pushed. Given this description, we can now further distinguish between intrusive interruptions and non-intrusive ones.

Generally, we consider that the intrusiveness of a notification displayed on a particular device depends on the preferences of the user and the context within which the notifications are received. Those interruptions that help the user or the user's group with the task at hand are not intrusions. Rather, they are *task support information* which we interpret as "good" interruptions. We define task support information as: *being related to another task (i.e. handling the content of the message) concurrent to the one being performed that will aid the latter's completion or enhance its efficiency.*

Thus, in information dissemination terms discussed above, we further classify intrusions and task support information as follows:

- intrusions are *unwanted (by the group or user) pushed information*;
- task support information is *pulled or useful pushed information (as determined by the user(s)'s preferences).*

Although some intrusive notifications might be unwanted by the group, they might nevertheless be considered important enough by the user receiving them for her to switch to handling the notification (i.e. disturb the group) rather than stick to the group task at hand (i.e. not disturb the group). This happens when there is a conflict of preferences between the group as a whole and the individual within the group. In such cases, the users would need to discuss whether the intrusion should be allowed or not. This would typically involve the users each stating their preferences regarding the intrusion in the current context and thus deciding as a group whether to allow the notification (i.e. whether they would mind the group being disturbed). However, if the

[1]We consider a specific aspect of interruptions here. However, an interruption may also be a request to take action on some issue. We will investigate this other aspect of interruptions in future work.

users do this themselves, the group task is necessarily disturbed. Moreover, the users may not want to reveal their true preferences about the notifications in case other users may want to exploit these preferences to display more notifications and therefore disrupt the former's attention repeatedly during the meeting. In some cases, the users might also want to delay the notification of messages of other users to a later point in time when the attention level required in the group activity is lower. In yet other situations, a user might allow other users to disturb the meeting if they agree to let the former receive messages at a later point in time during the meeting. This may happen if the former is expecting an important message or does not need to be particularly attentive later on during the meeting (e.g. if she has presented her work earlier during the meeting).

Given the above desiderata, we require an additional interface, between the notification controllers (i.e. the software that controls the notification devices) and the physical world. This interface is responsible for managing these complex interactions and resolving the conflicts over the decision to display incoming notifications. As discussed in chapter 1, negotiation is the main way of resolving such conflicts and, to this end, in section 8.3, we develop an agent-based mechanism that can flexibly negotiate the best course of action on behalf of the users. Before doing so, however, we detail in the next subsection the meeting room scenario.

8.2.3 Intrusiveness in the Meeting Room

The scenario involves a number of users meeting in a room that is fitted with pervasive computing artefacts that are fixed in the room (e.g. a smart whiteboard or an audio system capable of generating audio cues) or that are brought in by the users (e.g. laptops, PDAs, mobile phones) as can be seen on figure 8.1. The aim of the meeting is to discuss a group project which has a specific subject, and each user takes turns at voicing his viewpoint on the subject. The meeting may also involve presentations by group members on a particular issue of the project. Video calls are expected from other members who were not able to physically attend the meeting.

There are different ways a user in the meeting room can be disturbed. Here, we consider the following as the most relevant types of notification delivery services:

1. An email client — this device simply shows a header containing the email sender and subject (other details may be added but the content is not shown). This type of notification is intrusive to the extent that it alerts the user of the meta-information about the message rather than the content itself. This does not guarantee that the user will entirely shift his focus of attention to reading the email unless he finds the subject very interesting (i.e. negotiated interruption).
2. An instant messenger — this pops up a window and beeps the user. This type of notification gives the content of the message and disturbs the user's activity with

FIGURE 8.1: Intrusiveness in the meeting room. Users might be checking their email or sending SMS while attending a presentation, thus disturbing their colleagues.

the beep. This nearly always results in the user shifting his focus of attention (i.e. scheduled or immediate interruption).

3. A public display — this is a whiteboard that simply shows messages that are sent to it. This device is potentially the most intrusive since it disturbs the whole group as everyone in the meeting room is able to see the message. Users may re-route messages or video calls received on their laptops to this device whenever the messages are relevant to the whole group (i.e. scheduled or immediate interruption).

The participants of the meeting may reach different states of focus at different points in time. For example, in a presentation most users are focussed on the presentation, while if two users are in discussion, the others might lose focus altogether. In another context, the meeting might even be silent if all users are reading an important document together. The latter state would require a very high level of attention. At yet other times, the group might be having a coffee break which can allow intrusive notifications.

Given that each of the devices involves a particular degree of interruption (e.g. immediate as opposed to negotiated), it is possible to relate the preferences of users over a received notification or message to a given device through its degree of interruption. Thus, an important message to a particular user might be displayed on his IM, while an important message to the group should be displayed on the public device. However, when users have conflicting preferences regarding notifications, some form of negotiation is needed. To this end, the next section details our multi-agent based solution to negotiating and managing interruptions.

8.3 The Multi-Agent Solution

We have developed a multi-agent system for managing intrusiveness and have applied this system to a real meeting room (at our university). This system defers the handling of messages to software agents that each represent their owners. Specifically, we assume that users relinquish the decision about which device to use for a notification to their agent (after negotiations with other agents). This may mean re-routing an IM message to an email client or even being kept on an invisible queue for later (e.g. post-meeting) delivery depending on the preferences of the user. This is a fundamental change to the present situation in which the sender of the notification chooses the device on which his message will appear. To capture the group's influence on the display of a notification, we incorporate the use of a *dial* which can be turned up or down by the members of the group (with all members' consent) to regulate the *level of intrusiveness* allowed. Thus, at different points in the meeting, the users might want their agents to know that they do not want to be disturbed (except for very important messages) by turning the dial down. During a coffee break the dial can be turned up to signal to the agents that intrusions are allowed.[2] Here, we assume that the users have input their preferences into their representative agent to allow the latter to know which messages are to be considered important and which are not. Fundamentally, this involves assigning points (from 0 to 1 inclusive) to particular sender names and subjects that a notification could contain. For example, a sender named Wendy gets 1 point since she is the project supervisor and a subject such as 'Project guidelines' gets 1 point as well since it is relevant to the current meeting (about that project). On the other hand, sender names that are not expected or not deemed very important will get less than 1 point (including 0 expressing no interest in such notifications being routed to their target user).

8.3.1 Formal Definitions

The meeting room contains human users $h_1, h_2, ..., h_n \in H$ and devices $d_1, d_2, ..., d_n \in D$. There also exist other users outside the meeting room noted as $h'_1, h'_2..., h'_n \in H'$. Devices can have different characteristics: private display (OD) (e.g. email client), public display (PD) (e.g. the smart whiteboard) and part-private-part-public (POD) (e.g. IM) such that $POD \cup PD \cup OD = D$. Devices are controlled *indirectly* by user agents $\alpha, \beta, ... \in Ag$.

[2]While the dial is a manual means of managing the level of intrusiveness, we aim in the longer term to develop sensing devices to monitor the state of the meeting in order to adjust the level of intrusiveness automatically (e.g. by tracking the progress of the meeting through the agenda, by monitoring movements of users through a video processing tool to detect how users are interacting, or by assessing the level of noise to detect the level of interactivity between users).

By indirect control we mean that it is the system, a special user agent representing the meeting room and the *group of users*, called $SAgent$, that handles the actual display of messages, but it is the user agent that decides which device should be used. Thus, $SAgent$ carries out the display of notifications when asked by another user agent if the agent satisfies certain conditions. In this way, the $SAgent$ actually manages the group preference on the level of intrusiveness allowed. The behaviour of the $SAgent$ is regulated through the dial (controlled by the group of human users) which indirectly scales the level of intrusiveness of all devices in the meeting room by changing the conditions which $SAgent$ imposes on the display of notifications. Figure 8.2 shows the flow of messages between agents (including $SAgent$) whenever a notification is received from outside the meeting room. We generally capture the devices that are accessible by a user's agent by the function $G : H \rightarrow 2^D$. Each meeting room user can have a number of devices used to display (notifications of) messages. We use α_{h_1} to note an agent α belonging to user h_1. Devices under the indirect control of an agent are noted as $\langle \alpha, \langle d_1, d_2, ... \rangle \rangle$. Messages received by users from outside the meeting are noted as $m_1, m_2, ..., m_n \in M$. Each message has the following structure $m = \langle h', h, s, c, t, d \rangle$, where $h' \in H'$ is a sender outside the meeting room, $h \in H$ is the recipient inside the meeting room, s is the subject of the message, c is the content of the message, $t \in Time$ is the time at which the message arrives, and $d \in G(h)$ is (are) the device(s) available for display for that user. The meeting starts at time $t = 0$ and ends at a given time t_{end}.

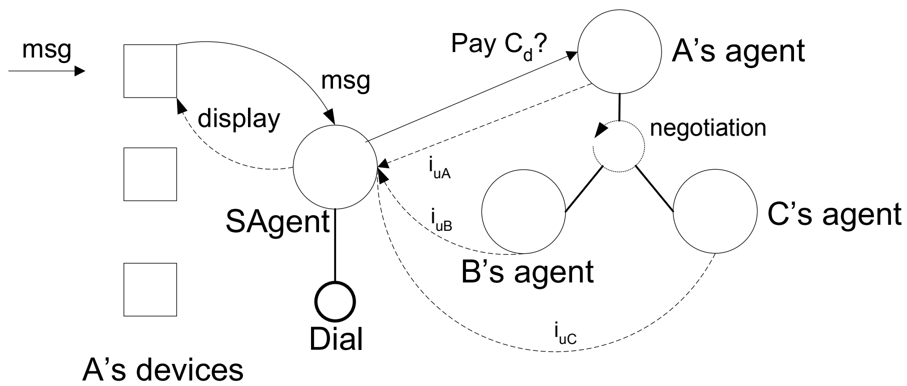


FIGURE 8.2: Interactions between device and agents. Dotted lines represent interactions after a message is received by user A's agent has negotiated with B's and C's agent.

We consider the intrusiveness to be a *cost* to the group activity since it disturbs the meeting; notifications can be allowed into the meeting if and only if the gain of displaying them matches the cost (or level of disturbance) to the group. The dial can be formalised as a function that scales the acceptable level of intrusiveness $K : 2^D \rightarrow [0, 1]$. Assuming each set of devices (i.e. POD, PD, OD) has a different degree of intrusiveness $Q \in [0, 1]$ (and therefore cost) in the following order $Q_{OD} > Q_{POD} > Q_{PD}$, then the actual cost of a particular device d in a particular context (as set by the dial), is obtained by the function $C_d : [0, 1] \times [0, 1] \rightarrow [0, 1]$ defined as $C_d = Q_{sd} \times K(d)$, where $sd \in$

$\{POD, PD, OD\}$. As can be seen in the function C_d, the dial scales the cost to display a message on each set of devices (private display, public display, part private-part public displays). We also assume the existence of an invisible queue that stores messages that are not sufficiently important to be displayed at a particular time, but which might become important enough later on. This device does not interrupt any user and is therefore assigned a cost of zero.

Agents negotiate about the display of a notification of a given message on a particular device. Also, agents negotiate about the particular point in time the notification is to happen. In this context, the negotiating agents may have different preferences about the message m. For example, some senders or subjects might be more preferred by one agent than another.[3] Agents may also have different preferences about the device chosen[4] d_{chosen} (e.g. a public device might be more preferred by an agent since it guarantees the alert will be perceived by the user while an email which needs to be polled is more preferred by the other agents since their owners are going to be less disturbed), and the time at which it is displayed $t_{display}$ (i.e. disturbing a user immediately may be preferred by an agent while other agents may prefer it to be displayed much later on at the end of the meeting). The content of the message is non-negotiable but the time of display and device chosen can be negotiated. All these issues are captured in a contract O such that $\{m_{received} = m, t_{display} = t, d_{chosen} = d\} = O$. For each of these issues, we assume the pairs of negotiating agents have linear utility functions U for each issue that follow the MMPD as described in chapter 4. The domain of values each issue takes is assumed to be finite (e.g. a fixed number of devices ranked in order of their level of guarantee to alert a user while the domain of time points is constituted of a fixed number of time points along the duration of the meeting).

The utility function applying over a contract O is then refined to capture the *points* assigned to a sender name and a subject, determine the utility of the device chosen and the time of display minus the cost (intrusiveness) of using that particular device as shown below:

$$U^{\alpha}(O) = \sum_{(x=v)\in O} w_x \cdot U^{\alpha}_x(v) - C_d \tag{8.1}$$

where $\sum w_x = 1$. Thus, the utility function returns a points obtained for having a given message displayed on a given device at a particular point in time. Whenever a message is displayed on a particular device, the *SAgent* rewards the agent concerned (i.e. the user agent which asked to display the message) with the number of points dictated by its utility function. This reward represents the user's reward to its agent for satisfying

[3] The sender name and subject are considered to be only the necessary rather than sufficient features of the utility function. The other elements of the message such as the content of the message (e.g. using data-mining techniques where possible) and the time at which it is received may also allow for a more comprehensive analysis of the utility of the message. We foresee doing so in future work.

[4] In contexts where agents cannot be trusted, the content might not be transmitted to the other agent when given for evaluation or some form of cryptographic technique used to encode the message and the utility function (see 3 for more details).

his preferences. The more points it gets, the more the agent is able to pay for messages that the user might like. Moreover, agents may also be allowed to exchange points they receive if they need to collaborate to pay for the cost of a message (whenever they cannot pay for a message on their own).

Messages from users, H, are received by the system agent $SAgent$ which manages the meeting room (i.e. devices forward incoming messages to the system agent and wait for a decision to be made before displaying anything). We assume that devices forward the messages they receive to the system to notify the user agent concerned. A message is first analysed by the system to determine the recipient h. The system then contacts the appropriate user agent α_h. The user agent then needs to make some decisions by taking into account the cost C_d of displaying a message m on the targetted device d, the time at which it wants the message displayed, and the utility of a message $U^{\alpha_h}(m)$ to itself. We assume that all agents are initially assigned a budget B^{α_h} equal to the cost of displaying a message for the most expensive set of devices (i.e. $B^{\alpha_h} = \max_{d \in G(h)}\{C_d\}$).

In this way, messages are first assessed using the preferences of the user and then the decision is made whether to ask the system to display or not. There are 3 possible courses of actions that a user agent can take:

1. Ask the system to queue the message, resulting in no cost.
2. Ask the system to display message by paying cost C_d with the budget B^{α_h} available. The more costly the device chosen is, the less will the budget of an agent be after a message is displayed and agents might prefer to display high value messages on the less costly devices to maintain a high budget. This is graphically shown in figure 8.3.
3. Ask other agents to contribute to pay the cost of displaying the message.

The first two options are straightforward to carry out. The agent simply needs to analyse the message and determine the payoffs. If the budget matches the cost of display and the payoffs will replenish (partly or fully) the budget, then the message is displayed. However, there might be cases where $B^{\alpha_h} < C_d$ and $U_{\alpha_h}(O) \geq 0$ meaning that the agent would get a higher payoff than C_d if it had the additional funds to match C_d. To be able to achieve this, an agent can therefore negotiate with other agents to get their contribution to the pool of funds and get the message displayed. The display can be on the users's private device or on other agents' devices (if they agree to this) or on a public device depending on the importance of the message to the user and the group. The user agent will therefore negotiate with other user agents for their investments to match the cost of displaying the message. These other agents might have an interest in getting the message displayed since they might also have a preference for the sender and the subject. Agents may also have different preferences about the time of display and the

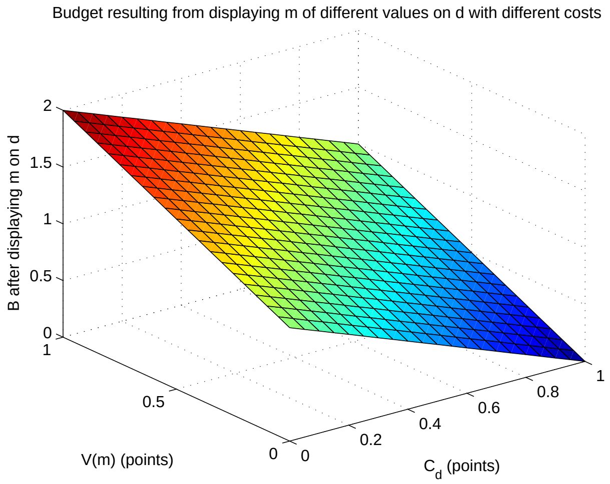


FIGURE 8.3: The budget B resulting from displaying a message m on a device d for different costs and values of the message displayed

device chosen such that it is possible to propose alternative devices on which a message can be displayed and times at which these events can happen. Moreover, since the agents are assumed to be selfish, we can expect that they might renege on their commitments to display a message on a particular device in order to increase their utility. Given this, in the next section, we describe the negotiation mechanism that aims to resolve the agents' conflicting preferences while taking into account their trustworthiness.

8.3.2 Persuasive Negotiation

In order to allow agents to reach efficient agreements quickly, we choose the persuasive negotiation model described in chapter 7. We choose such an approach specifically because it incorporates the use of promises of future rewards (i.e. trading points) together with proposals exchanged (e.g. display on IM and promise to give 0.3 points in the next encounter if the opponent agrees to 0.3 points now). Thus, not only can agents negotiate about the type of device to display a message on, but they can also promise points to each other in order to get their proposal accepted. Given that such promises are more likely to persuade the opponent (since they obtain points in return or have committed to give some) to agree to contribute to the payment than only proposing a device and time of display, the negotiating agents are expected to find an agreement more quickly than if they operated without such promises (as shown in chapter 7). Moreover, given that the agents are likely to meet repeatedly during the meeting as notifications are received, the use of promises allows the system to flexibly deal with important messages (to a user or the group) over time such that important messages are not rejected simply because the user's budget size varies as his messages are notified.

We will also assume, in this scenario, that agents are selfish as users might not want to be disturbed by messages of other users. In this case, the agents might not want to give all the points they promise or display the notification on a chosen device (in order to get more points). Given this, agents can use the CREDIT trust model to shrink or expand their negotiation ranges over the time of display and the device chosen. For example, as in CREDIT, a high level of trust would equate to agreeing to the proposed device and time of display while a low level of trust would equate to negotiating for later time of display and less intrusive device. Moreover, by using CREDIT, agents may also modify the promises they make according to the trust they have in their opponent. A high degree of trust would equate to a reasonable demand for extra points, while a low degree would equate to a high amount of points demanded since the agent would be expected to defect on the number of points provided.

Next we describe the algorithm used by the agents to perform negotiations with other agents in their environmentand decide which device to choose for display.

8.3.3 The Negotiation Algorithm

The algorithm is described in figure 8.4. Note that $-h \equiv H/h$ and $Tell_{\alpha \to \beta}(x)$ (for α tells β about x) is a message from α to β with the content x. From Steps 1 to 6 the agent determines its utility for all possible offers that it can make, sends the different proposals and arguments with them according to its level of trust previously calculated (using a trust model such as CREDIT). The type of argument (i.e. asked or given) and the offer is determined according to the RBT algorithm presented in section 7.3.5. In this case, given that the proponent is involved in multi-party negotiations and might make rewards or ask for rewards over multiple sequential negotiations with different parties, the value of the reward α_h can give or ask for in each encounter with other agents is at most equal to:

$$\max\{0, U_{\alpha_h}(O) + B^{\alpha_h} - \sum i_h^*\}$$

where i_h^* represents the points *promised or asked for the current offer* O by agents other than α_{-h}. In so doing, α_h makes a promise to α_{-h} that it will be able to refund after it can get its message displayed. In Step 5 each agent sends its potential investment i_{-h} for the offer given. Step 8 computes the pool of points available given the utility to be obtained and the cost of the device. Then, in Step 9, α_h selects the device and the time of display for which it gets the maximum investment and checks whether the points to be obtained are greater than zero. If α_h does get some points, in Step 11 to 14 it notifies all other agents about its decision so that they can update their commitments (i.e. keep track of promises) and send their investments to *SAgent* which updates their budgets (i.e. by deducting the promised i_{-u} for the current message), otherwise it queues the message. In Step 16 α_h forwards the payment for the device to *SAgent*. In Step 17, the *SAgent* pays α_h and Step 18 updates α_h's budget.

Require: $Trust(\alpha_h, \alpha_{-h})$
for all α_{-h} **do**
  use $Trust(\alpha_h, \alpha_{-h})$ to adjust negotiation range in offer O.
  while α_{-h} and α_h disagree **do**
    $Tell_{\alpha_h \rightarrow \alpha_{-h}}(illoc(O, i^*_{-h}))$ where $illoc \in \{propose, reward, askreward\}$ % offer O and, if possible, argument i^*_{-h} to other agents using RGM and CREDIT.
    $Tell_{\alpha_{-h} \rightarrow \alpha_h}(i_{-h}, (O, i^*_{-h}))$ where $i_{-h} \leq B^{\alpha_{-h}}$ and $i_{-h} \in [0, 1]$ % agents return their promise of contribution for the pair (O, i^*_{-h}).
  end while
end for
$\forall O \in \mathcal{O}$ calculate $Sum_i^O = \left\{\sum_{h \in H} i_{-h}\right\}$ % sum investments promised for each offer.
$O_{\max} = \arg\max_{O \in \mathcal{O}} \left\{U^{\alpha_u}(O) + Sum_i^O\right\}$ % choose utility maximising offer.
if $B^{\alpha_h} - C_d + Sum_i^O > 0$ **then** % if a device and time of display can indeed be chosen
  for all α_{-h} **do**
    $Tell_{\alpha_h \rightarrow \alpha_{-h}}(O_{\max})$ % tell other agents the choice.
    $Tell_{\alpha_{-h} \rightarrow \alpha_h}(i_{-h}, O_{\max})$ % agents reply to α_h with their investments. This is where agents can defect on payments.
    Update $Trust(\alpha_h, \alpha_{-h})$ % update the trust model.
  end for
  α calculates its own investment $i_h = \min\{B^{\alpha_h}, C_d - Sum_i^{O_{\max}}\}$
  $Tell_{\alpha_h \rightarrow SAgent}(i_h, O_{\max})$ % give investment and contract chosento $SAgent$.
  Update trust of other agents % Using CREDIT here.
  $Tell_{SAgent \rightarrow \alpha_h}(points = U^{\alpha_h}(O_{\max}) + Sum_i^{O_{\max}})$ % $SAgent$ gives points to α_h including any extra points from investments.
  $B^{\alpha_h} = U^{\alpha_h}(O_{\max}) + Sum_i^{O_{\max}} + B^{\alpha_h} - i_h$ % agent updates its budget with new points.
else
  send to queue
end if

FIGURE 8.4: Algorithm to determine most appropriate device and time to display message

8.4 Implementation

In order to evaluate the efficiency and effectiveness of our algorithm we developed it using the Jabber[5] platform (a highly extensible instant messaging system). In more detail, the Jabber platform incorporates devices and agents in the following ways:

1. Devices with various levels of intrusiveness are represented by a number of highly configurable instant messaging clients. Thus, Jabber clients (e.g. Psi) can be configured to simulate an email client by having messages sent to a client that simply displays an icon when a message is received. The client then needs to be checked (or polled) to view the message. An IM can instead be simulated by having the client pop up a chat window and beeping at the same time. Thus the user is alerted and the message can be viewed immediately. Other devices mentioned as part of the meeting room, such as the public device and the invisible queue, can be created by having a custom-made Jabber client that simply outputs messages it receives to a window and an internal queue that is not visible respectively.

[5] `http://jabber.org/`

2. Software agents as pictured in figure 8.2 are made to interact on a server that plugs into the Jabber system that is responsible for routing XML-based messages which come from users outside the meeting room. In this way the negotiation is performed in a single thread of control every time a message is received. Thus, after negotiation, agents can provide the appropriate routing information to the Jabber system (i.e. which device to be chosen for notification).

In the next section, we detail the operation of our system.

8.4.1 System Operation

Each user in the environment is assigned their own unique 'Jabber ID'. Associated with that identifier there are a number of resources, in the scenario's case there are two; an e-mail 'device' and one for instant messages. The other two candidate devices for notification delivery are the invisible queue device, and the public whiteboard display (a first class Jabber ID in its own right), shared amongst all of the users in the scenario. The various components of our system are shown in figure 8.5.

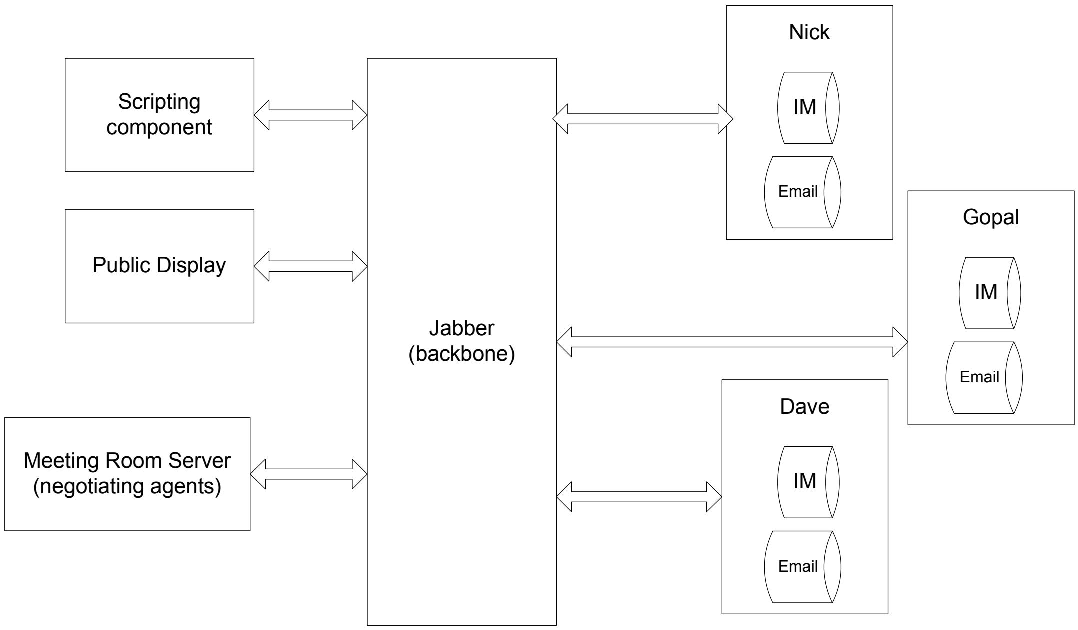


Figure 8.5: The architecture used for the meeting room and negotiating agents.

The user agents were implemented within a Jabber server component (i.e the meeting room server), representing a meeting room. The meeting room server maintains an internal description of each user's preferences as part of the Jabber system's user profile (i.e. U^{α_h}). The user can view or change his preferences via dialogue (using an instant messenger) with this component. This preference information is then used to initialize the user's agent, which is created when the user first logs in to the system. As a user adds further devices to the system, this agent is then informed of the new device, and thus different components become candidate targets for notifications.

Messages that are sent from outside the meeting to a particular Jabber user go through the Jabber server, which then re-routes them to the meeting room server (which represents the *SAgent*). The agent representing the recipient is then notified of the message so that it may begin to negotiate for an appropriate display device. The meeting room server receives the resulting choice of the agent and provides the Jabber server with the appropriate routing information.

To illustrate the operation of our system, consider the following interaction episode. We will assume that Nick, Gopal, and Dave are having a meeting. Each user has a Psi-based email client and an IM client up and running on his laptop while the public display client is connected to a smart whiteboard and the invisible queue is implemented in the meeting room server. Before the users start the meeting, they log on to the Jabber system which communicates their presents to the meeting room server. The latter then queries the users for their preferences. The meeting topic is about "FEEL project" which all users register in their preferences in their profile (e.g. they each give 1 point to that subject to indicate a high preference). Moreover, they each assign, possibly different, preferences for senders (e.g. Nick gives Wendy 1 since she is his boss while Dave gives Wendy 0.5 since she is not involved with Dave on any projects at the moment) and other subjects including the meeting subject. Duplicate entries are prevented by the system. Let us assume in the following that a message (e.g. an email) is sent to Nick by Wendy about the meeting subject in particular and that the dial is set to $K(d) = 1$ such that a message to a public display would cost 2.5, an IM 2.2, an email client 1.0 and the invisible queue 0, and that each agent is given an initial budget of $B^{\alpha_h} = 1$:

1. The meeting room server (i.e. *SAgent*) intercepts a message 'from' `wendy@scenario` with subject `FEEL project` to recipient Jabber ID of Nick, `nick@agentbox.scenario`.

2. The *SAgent* dispatches the message (including message metadata contained in envelope) to the agent representing the interests of the target user (i.e. α_{nick}).

3. α_{nick} first calculates the utility of the message using equation 8.1 and then negotiates with α_{dave} and α_{gopal} as per the algorithm described in figure 8.4. As can be deduced from our initial settings, α_{nick} can only afford an email or invisible queue by itself but given investments of other agents, it could send the message to the public device or the IM. Given that Gopal and Dave have a high preference for the meeting room subject and that Dave also has a high preference for the sender while Gopal has none (i.e. from equation 8.1, $U^{\alpha_{gopal}} > 0$ and $U^{\alpha_{dave}} > 0$), each decides to invest different amounts in the message for different devices that could be used for the notification. Let us assume (according to preset values of P_D)

that the utility maximising device (without promises) for α_{nick} is the IM which attracts an investment of $i_{gopal} = 0.2$ from Gopal and $i_{dave} = 0.4$ from Dave. Instead, with a promise of returning $i^*_{Gopal} = 0.1$ to Gopal and $i^*_{dave} = 0.1$ to Dave, the utility maximising option for α_{nick} is when it uses a public device. Thus α_{nick} can get $i_{gopal} = 0.8$ from Gopal and $i_{dave} = 1.0$ from Dave's agent for the public display (for which they would invest more than the IM without the promises but these investments would not be enough to satisfy the cost of the public device). Nick's agent can thus display the message on the public display by investing only $i_{nick} = 0.9$ and rewarding α_{gopal} and α_{dave} in future encounters.

4. α_{nick} sends the identifier of the chosen device to *SAgent* together with the investments of all agents.

5. The *SAgent* then sends the whole content of the XML-based message from Wendy to the Jabber system with the appropriate routing information that selects the public whiteboard.

6. The *SAgent* then rewards all the agents with the utility they gain from the display of the message on the public device (i.e. 2 to α_{nick}, 1.5 to α_{dave} and 1 to α_{gopal}).

8.5 Summary

In this chapter, we have presented an agent-based system to manage intrusiveness in pervasive computing environments. The solution takes into account the preferences of a user, and other users in his environment through our model of PN and CREDIT, in deciding the intrusive level of a message. Moreover, we successfully implemented the algorithm in Jabber and deployed it in a meeting room scenario. Thus we have achieved our main objectives towards showing the applicability of our negotiation models to solving conflicts prone to uncertainty in practical applications (see section 1.5).

In general, the main findings of this work were that the multi-agent negotiation algorithm would always choose the most important incoming messages for display and, if too many messages of medium importance are received, the agents gradually run out of budget and cannot afford to display any further messages. This results from the relationship between the budget and the value the agent obtains from the display of messages on certain devices as shown in figure 8.3. The more costly the device, the lower the resulting budget after a notification, hence the potential of an agent to display notification next time decreases. Moreover, the negotiation algorithm allows agents to adapt their behaviour over time, through the use of arguments and trust, to permit important notifications when their budgets are low and reduce their contributions to untrustworthy agents respectively.

Chapter 9

Conclusions

The various models we have developed in this thesis are linked by the underlying theme of attempting to reduce the uncertainty in negotiations in multi-agent systems. Therefore, in this chapter we bring together the main achievements of these models and discuss how they impact on the wider issues that pervade the field of multi-agent systems.

The rest of this chapter is structured as follows. Section 9.1 summarises the main results of this thesis, while Sections 9.2 and 9.3 discuss the theoretical and practical implications of our work respectively. Finally, Section 9.4 discusses future lines of work concerned with reducing uncertainty in multi-agent negotiations.

9.1 Summary of Results

Our main aim at the beginning of this thesis was to develop mechanisms that would enable the resolution of conflicts under uncertainty. In particular, we set out to devise techniques to reduce uncertainties about the agents' reliability, honesty, preferences, and action sets, when agents are involved in negotiation. We applied our techniques to the two main classes of negotiation mechanisms, namely mechanism design and bargaining. In both cases, these techniques were based around the notions of trust and persuasive negotiation.

Thus, using mechanism design principles, we developed the area of Trust-Based Mechanism Design. This aims to produce efficient solutions by reducing the uncertainty about the agents' preferences through a protocol and uncertainty about their reliability through the use of trust. Thus, our Trust-Based Mechanism is the first reputation mechanism that is incentive compatible, individually rational, and efficient. Moreover, it was shown that our TBM can use any trust model to produce efficient outcomes in the long run, as the trustworthiness (reliability) of all agents are learnt over multiple interactions by the trust model.

In the area of bargaining, we aimed to reduce uncertainty about agents' honesty and reliability through the use of a trust model that could accordingly adjust the agents' negotiation stance. To this end, we developed the CREDIT trust model. This is the first such model that can reduce uncertainties in bargaining encounters. Specifically, in CREDIT we showed how trust, learnt over multiple interactions, could be used both to constrain the domains of issues being negotiated and choose issues to be negotiated. In so doing, CREDIT is able to avoid exploitation by unreliable and dishonest agents. CREDIT was also shown to elicit profitable outcomes against agents that are reliable to a certain degree.

Given that CREDIT only reduces uncertainty regarding the reliability and honesty of agents, we developed a novel model of Persuasive Negotiation that reduces the uncertainties about the agents' preferences and action sets in bargaining encounters. Thus, we provided a new protocol for PN that takes into account rewards that can be asked from or given to another agent. This protocol reduces the uncertainty about the type of actions agents are allowed to perform. In so doing, we also provided the first protocol that clearly specifies the main commitments agents make when engaging in persuasive negotiation. Moreover, we provided a novel decision making model for agents engaging in persuasive negotiation. Thus, we provided a mechanism that generated arguments in the form of rewards that constrain the outcome of repeated encounters (i.e. that constrain the agents' action sets). These arguments try to give more value to an offer (than the offer by itself) on the present encounter by providing guarantees on the outcome of future encounters and therefore speed up the search for an agreement (without knowing the opponents' specific preferences). Thus, we showed that through persuasive negotiation agents are able to reach agreements in less time than in the non-persuasive case and obtain a higher overall utility. Furthermore, we developed a new strategy for persuasive negotiation that selects the offers and rewards that are most likely to persuade an opponent and maximise the agents' utilities over repeated encounters.

Finally, we provided an example application of CREDIT and our model of PN through the model developed to manage intrusiveness in pervasive computing environments. Thus we showed how the intrusiveness of notifications in a pervasive computing environment could be reduced by allowing agents to negotiate, on behalf of their owners, the display of these notifications. In so doing, we provided the first practical application of both PN and trust in multi-agent negotiation.

In short, the models we have described above form part of a wider initiative to solve the problem of uncertainties in multi-agent interactions. In the next section we discuss their theoretical implications for research in multi-agent systems in general.

9.2 Theoretical Implications

Conflict resolution in multi-agent systems is a major issue that has always received a significant degree of attention in the agent-based research community. In particular, mechanism design and bargaining have been at the centre of this endeavour. This effort has lead to a number of models that each aim to elicit predictable and efficient outcomes given certain constraints. In this context, the work presented in this thesis has tried to reduce the constraints on these mechanisms so as to make the solutions more widely applicable in realistic settings.

Against this background, CREDIT represents a first attempt at using trust in automated negotiation. In CREDIT, trust is used both to choose issues to be negotiated and their corresponding negotiation ranges. These uses of trust borrow ideas from the human negotiation literature such as Fisher and Ury (1983) and Raiffa (1982). Thus, our work on CREDIT has showed that trust can make or break relationships between agents as they do in the human domain (Gambetta, 1998). Specifically, trust can either widen negotiation ranges and allow for more profitable agreements in the long run or it can shrink negotiation ranges so as to recover previous losses or reduce the risk of losing utility in an encounter. In so doing, CREDIT's shrinkage of negotiation ranges is synonymous to increasing an agent's bargaining power since this procedure results in higher utility for the agent (than without the shrinkage) if an agreement is reached. Obviously, reducing the negotiation ranges also reduces the probability of reaching an agreement as the negotiation ranges of the agents may not intersect anymore (which happens when interacting with nasty agents). Nevertheless, this reduction of negotiation ranges is sometimes useful since it serves to avoid unreliable and dishonest agents.

In general, through CREDIT, we have provided the first insight into procedures that allow agents to specify their negotiation ranges according to the known characteristics of the opponent they encounter. Previously, this was not possible and heuristics for negotiations relied on a rule of thumb to specify negotiation ranges for the agents as in (Faratin et al., 1998; Fatima et al., 2001). Moreover, in CREDIT we have shown how societal factors can impact on automated negotiation. By introducing aspects such as institutions and norms that could impact on trust, CREDIT can adapt to the context within which it is used (and therefore adapt the negotiation stance of an agent accordingly). Up to now, negotiation models had hardly assessed the impact of such societal factors on the outcome of negotiations.

The social aspect of interactions has also been neglected in mechanism design up to now. Indeed, mechanism design relies on micro-economic principles that tend to boil all the attributes of the agents down to what is termed their 'type'. For example, the different degrees of reliability of an agent could be defined according to different types (Ely et al., 2004) or the value an agent attributes to a particular good is also usually defined by its type. While the analysis resulting from such a modelling technique is rigorous and precise, it assumes that all possible types of agents are known apriori. However, this is not the case in most realistic applications, where for example, a mechanic's reliability is only known after a number of interactions with him or a buyer's valuation of some goods may only be known (by herself) after analysing the quality of the goods or the need for the goods. Our work in TBMD captures such aspects of agents which have been usually avoided by game theoretic models. TBMD achieves this by separating the type of an agent from the reliability other agents attribute to it through their trust model. The latter relies on the agents' information gathering capability to output the believed reliability of another agent. Therefore, as was shown in chapter 6, the mechanism perfects its outcome as the agents refine their measure of trust over repeated encounters. Through TBMD, we provide the first mechanism that connects mechanism design to the social aspects of an agent since the trust model (as was shown for CREDIT) can capture many of the social attributes that impact on an agent's decision making. For example, the reputation other agents have in the society or the agents' similarity in their assessment of others may determine how an agent perceives the reliability of another agent.

In general, TBMD differentiates itself from current mechanisms by the fact that it generates the efficient outcome (resulting in maximum profit and choosing the most reliable agents) after a number of encounters rather than in one shot. The repetitive aspect of TBMD is needed in order to remove the assumption that all agents are believed by other agents to be completely reliable. Thus, through TBMD as well as CREDIT, it is expected that, in the long run, only the most reliable and trustworthy agents will survive in a population as the unreliable ones are avoided and cannot make any profit. In so doing, these mechanisms may neglect the fact that agents may have varying reliabilities. Thus, over time, agents could either get better or worse depending on circumstances that may or may not be known a priori. For example, TBMD would avoid unreliable agents and not select them in a future encounter where they could have been more reliable. CREDIT reduces the possibility of overlooking a reliable agent by leaving the negotiation mechanism to decide the fate of a previously unreliable agent. However, this does not take into account the fact that an agent may know its opponent is going to be more or less reliable in future. If it did, the agent could shrink negotiations in the encounter where the agent is more reliable (and claim more utility) and expect lower profits when the agent is less reliable (and hence negotiate with more relaxed ranges). This could also help the agent make other parallel decisions more efficiently. In general, communicating information other than the costs and valuations of an agent to another falls into the realm of ABN (as discussed in chapter 2). In this thesis we developed a particular aspect of ABN through our PN model.

Using PN we have shown how agents can use arguments to influence repeated encounters positively. As we highlighted in chapter 1, there are very few negotiation models that allow agents to influence repeated encounters as we do in PN. Moreover, there are also very few ABN mechanisms that have been applied to solve the particular types of conflict that arise in multi-agent systems. Given this, we believe repeated bargaining encounters could be used as a testbed for ABN mechanisms since they allow them to benchmark their properties directly against other basic negotiation tactics as we have done in this thesis (see chapter 7). ABN mechanisms may do so by specifying arguments in repeated bargains which consider operations on the negotiation object that is negotiated over in each encounter. As we have shown in this thesis it is then straightforward to specify arguments that the agents can directly evaluate.

9.3 Practical Implications

Having discussed the theoretical implications of our models in the previous section, we turn to their practical implications for multi-agent systems in general. Specifically, as we propose in chapter 3, we believe the semantic web provides agent researchers with many possibilities for applying their work to practical applications. The Grid, peer-to-peer systems, and pervasive computing environments, are yet further fertile areas that share similar issues with the semantic web and therefore, we believe are likely to make use of the variety of models developed for multi-agent systems. In all these domains, we believe the management of resources will be handled by intelligent agents which can autonomously choose their interaction partners and negotiate with them. These systems are all prone to the uncertainties we have considered in this thesis (see chapter 1) which make our models of trust and persuasion particularly suitable for them.

For example, the particular need for trust in such applications has been recognised by the semantic web community which places trust at the top of the semantic web 'layer cake' (see figure 9.1). The fact is, and this constitutes the thrust of this thesis, that trust underlies all interactions prone to uncertainty and such uncertainty pervades all interactions that are performed over unsupervised and open systems such as the semantic web or the Grid. As a specific example, it is possible to concretely apply TBMD in running online auctions where buyers and sellers are allowed to state their trust in each other when negotiating over resources that are available in the Grid or the semantic web (see appendix B for a worked example). This has become possible thanks to the work by Giovannucci et al. (Giovannucci et al., 2004) who have developed an agent-based online (combinatorial) auction mechanism, iBundler, that is currently being integrated with TBMD. This combination will allow auctions to be more flexible and adapt to richer information (as opposed to the only use of cost and valuations) in making allocations. This, we believe, will also benefit e-business (Sadeh, 2002) at large because it encourages trustworthy behaviour in sellers and buyers who use the system to trade.

Where centralised systems such as TBMD cannot be applied, CREDIT could be used to allow agents to negotiate resources without the need for a central auctioneer when

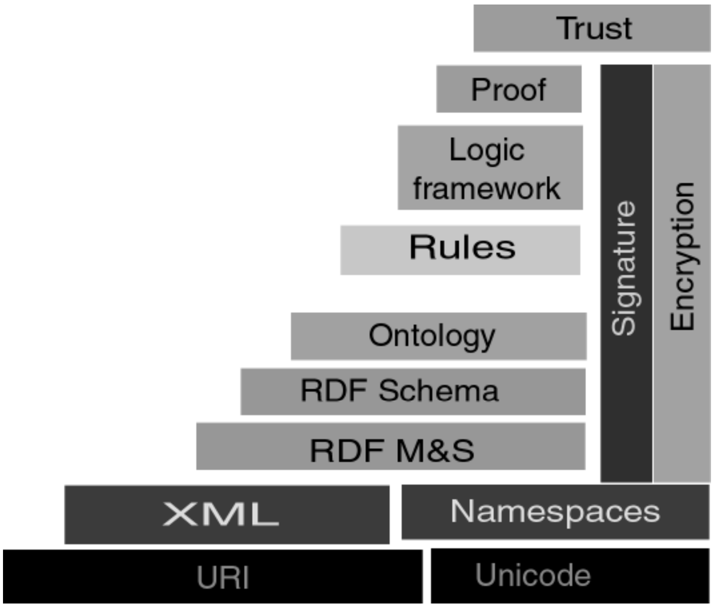


FIGURE 9.1: The semantic web layer cake proposed by Berners Lee (at www.w3.org /2002/Talks/ 04-sweb/ slide12-0.html).

the agents' reliability and honesty are prone to uncertainty. Thus, for example, agents that share files in peer-to-peer systems could avoid agents that free-ride over their resources by directly negotiating downloads with them according to their trustworthiness in providing the files they possess (Feldman et al., 2004) (i.e. taking into account the quality or download speed for example). Combined with a reputation model (such as REGRET or Yu et al.'s), agents using CREDIT could then spread the trustworthiness of their counterparts throughout the network and prevent free-riders from exploiting other agents.

The other potential application area of agent-based systems we have explored in this thesis is that of pervasive computing environments. In this domain, the problem of managing intrusiveness has largely been neglected by community. Rather researchers have focused on using agents to perform identification, authentication, and perform transactions or to transmit information (e.g. instant messengers, or chat room bots) (Satyanarayanan, 2001; Schmeck et al., 2002). Moreover, most applications of pervasive computing consider mostly cooperative settings (e.g., multi-sensor networks (Manyika and Durrant-Whyte, 1997) or sharing information through smart phones (Islam, 2004)). In contrast, our work presented in chapter 8 presents a novel way of developing pervasive computing applications in competitive settings by using agents to manage the preferences of the users in a dynamic fashion through negotiation rather than through constraints satisfaction algorithms used in the cooperative case. Also we highlight the use of agents in group applications where the need for focus is important and this focus needs to be adjusted according to the context and the conflicting preferences of the users as individuals and the group as a whole. Specifically, our persuasive negotiation mechanism goes some way towards solving this competitive side of interactions in pervasive computing environments. Moreover, through our PN model, we have shown how agents, using arguments, can partition resources more efficiently over time by allowing important messages to be displayed even if agents run out of budget.

Our PN mechanism could also be applied in peer-to-peer systems or Grid-based applications where agents need to repeatedly and autonomously negotiate over the partition of resources such as computing power or storage space. Thus, through the use of rewards, agents can avoid losing customers when they are heavily loaded (i.e. having many users at the same time) by negotiating for a certain level of service commensurate with their capability to deliver the service and promising rewards on future contracts that may be made. Conversely, agents could ask for future rewards when accepting a lower level of service from a particular agent. In this way, the resources distributed over the system can be more efficiently used over time. In so doing, the system of self-interested agents can achieve a level of efficiency close to that of a cooperative group of agents.

9.4 Open Challenges

The work presented in this thesis is a step towards engineering robust and efficient protocols and reasoning mechanisms for open multi-agent systems prone to uncertainty. While we have considered issues that are prone to uncertainty which may affect the outcome of negotiations such as the reliability, honesty, preferences, and action sets of agents, there remains a number of other important challenges that need to be met for automated negotiation to be more robust to uncertainty. In particular, the automated negotiation mechanisms need to be able to handle uncertainty about the efficiency of the communication mechanism used and the computational capability of agents.

The efficiency of communication mechanism is determined by, amongst other things (e.g. by the noise in the information transmitted or by the size of bandwidth available), the time lag it allows between offers that agents send to each other. If the communication mechanism is not efficient (i.e. there is a long time lag) and the agents' environment is very dynamic (as we expect it to be in open distributed systems), the agents' preferences may change when offers are delayed. Hence, the inferences of one agent about its opponent's preferences may be completely wrong and reduce the attractiveness of offers or rewards made using these inferences. Moreover, agents may also find offers sent by their opponent less attractive than they were at the time they were originally sent. In such cases, the agents may end up taking a long time to find an agreement that is likely to be sub-optimal. Therefore, techniques must be devised to cope with the dynamic nature of the agents' preferences in the negotiation to allow agents to come to good agreements. This could be achieved by devising agents' strategies according to the dynamic features their preferences and devising a negotiation protocol that takes such factors into account.

Another major factor in negotiations that is prone to uncertainty is the computational capability of agents. The computational capability of an agent determines its ability to compute good offers in a timely fashion. This ability may be limited to such an extent that it is not possible to calculate the optimal offers within the time allowed by the protocol. This is because agents are likely to have combinatorial valuations about the issues being negotiated and may impose additional constraints on the values these issues take, which make the generation or evaluation of offers computationally expensive (Pekec and Rothkopf, 2003). Given this, agents may not be able to generate the most preferable offer (to them) or find an agreement that meets their opponents' constraints and combinatorial preferences. Therefore, we believe that agents' strategies must be designed to allow fast evaluation and generation of combinatorial offers and adapt to an opponent's computational capability (e.g., by using arguments to show an opponent that some constraints cannot be satisfied by the offers received from it). Also, protocols could be engineered to reduce the computational complexity of evaluating such offers.

We also believe that our work opens up a number of further possibilities in the particular areas of trust and ABN. We will first consider the challenges that still need to be met in the area of trust:

Collusion Detection — very few existing reputation or interaction mechanisms can prevent or deal with collusion (Sen and Sajja, 2002; Brandt, 2002). Moreover, while we have shown how agents can learn to reciprocate good actions over time, it has not been shown how they could learn to collude, which is equivalent to reciprocating to only some agents and sharing false information about these accomplices to exploit others. There are clear benefits to collusion as highlighted by Conitzer and Sandholm (2004), and we can therefore expect agents to collude in an open environment whenever this is possible. If the system is to be robust and incentive compatible, collusion should be prevented either through the application of a certain protocol (through mechanism design) or at the level of trust models which try to recognise colluders. Otherwise, agents could end up wrongly trusting others that are, in fact, exploiting them.

Social Networks — while most reputation models or security mechanisms (to some extent) assume that there exists a social network, the connections between the nodes in the network are rarely, if at all, given a meaning. That is, the semantics of connections are not detailed. Connections have mostly been used to represent past interactions among the agents in the community (i.e. a connection means that an interaction has occurred between the two nodes at its ends) or are simply given to the agents (Sabater and Sierra, 2002; Yu and Singh, 2002b; Schillo et al., 2000). A clearer definition of relationships (e.g. as collaborators, partnerships in coalitions, or members of the same organisations), defining the connections within the network would be needed in order to make trust models practically applicable.

In the area of ABN the following issues still need to be addressed:

Engineering Efficient Protocols — there are a number of protocols, including the one presented in chapter 7, which aim to precisely determine the commitments resulting from the illocutions made and the participation rules of the negotiation. While most of these protocols have been engineered to ensure termination of the negotiation dialogue or dictate the exact allowable moves of the participants, they have rarely been engineered to ensure specific outcomes. Moreover, most game theoretic models of bargaining only analyse existing protocols of bargaining (Muthoo, 1999) rather than trying to develop new protocols that ensure that the strategies available to the agents will result in an outcome close or equal to the efficient partitioning of resources. It would therefore be a significant step forward if bargaining protocols were developed to allow agents to negotiate in a distributed fashion and ensure efficient outcomes are selected.

Preferences — as shown in chapter 2, ABN aims to provide a mechanism to change preferences of agents during negotiations by providing justifications. In collaborative settings this is easier since the agents can totally trust each other and assess the information given to make further decisions. However, when agents are selfish, the information and justifications they give may only be such that they result in a higher utility for the agent sending them. In such settings, agents may need to verify the information transmitted or rely on their trust in their opponent to accept such information. Moreover, if agents can autonomously change their preferences according to new information received from other agents, their human owners may not obtain what they specified as their preferences to their agent. Therefore, more work needs to be done to ensure that agents can indeed exchange arguments that can convince other self-interested agents to change their preferences during negotiation and make sure that these changes are still agreeable to the agents' owners. This will ensure the predictability and robustness of the system.

The advent of such technologies as the Grid, semantic web, and pervasive computing, has widened the scope of potential applications of MAS, as well as the range of issues MAS researchers have to consider in developing their systems. It is therefore crucial that the challenges we have identified here be met to ensure that MAS are secure, efficient, and result in profitable outcomes for their users so as to be applicable in a wide variety of domains.

Bibliography

G. D. Abowd and E. D. Mynatt. Charting past, present, and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 7 (1):29–58, 2000.

C. Adams and S. Farrel. Rfc2510 - internet x.509 public key infrastructure certificate management protocols. Technical report, The Internet Society, http://www.cis.ohio-state.edu/htbin/rfc/rfc2510.html, 1999.

L. Amgoud and S. Kaci. On the generation of bipolar goals in argumentation-based negotiation. In I. Rahwan, P. Moraitis, and C. Reed, editors, *1st Intl. Workshop on Argumentation in Multi-agent systems*, 2004.

L. Amgoud and N. Maudet. Strategical considerations for argumentative agents (preliminary report). In S. Benferhat and E. Giunchiglia, editors, *9th International Workshop on Non-Monotonic Reasoning (NMR 2002): Special session on Argument, Dialogue and Decision*, pages 399–407, 2002.

L. Amgoud, N. Maudet, and S. Parsons. Modelling dialogues using argumentation. In E. Durfee, editor, *Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS 1998)*, pages 31–38, Boston, MA, 2000a. IEEE.

L. Amgoud, N. Maudet, and S. Parsons. An argumentation-based semantics for agent communication languages. In F. V. Harmelen, editor, *Proceedings of the European Conference on Artificial Intelligence (ECAI-2002)*, pages 38–42, Lyon, France, July 2002. IOS Press.

L. Amgoud and S. Parsons. Agent dialogues with conflicting preferences. In J.-J. Meyer and M. Tambe, editors, *Proceedings of the 8th International Workshop on Agent Theories Architectures and Languages (ATAL 2001)*, pages 190–205, Seattle, 2001.

L. Amgoud, S. Parsons, and N. Maudet. Arguments, dialogue, and negotiation. In W. Horn, editor, *Proceedings of the European Conference on Artificial Intelligence (ECAI 2000)*, pages 338–342. IOS Press, 2000b.

Aristotle. *Topics*. Clarendon, Oxford, UK, 1928. (W. D. Ross, Editor).

R. Ashri, I. Rahwan, and M. Luck. Architectures for negotiating agents. In V. Marik, J. Müller, and M. Pechoucek, editors, *Multi-Agent Systems and Applications III: Proceedings of the 3rd International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS 2003)*, volume 2691 of *Lecture Notes in Artificial Intelligence*, pages 136–146. Springer Verlag, Berlin, Germany, 2003.

J. L. Austin. *How to Do Things with Words*. Harvard University Press, 1975.

R. Axelrod. *The Evolution of Cooperation*. Basic Books, New York, USA, 1984.

B. Banerjee and J. Peng. Countering deception in multiagent reinforcement learning. In *Proceedings of the 6th workshop on Trust, privacy, deception and fraud in agent systems*, pages 1–5, 2004.

R. Becker, S. Zilberstein, V. Lesser, and C. V. Goldman. Transition independent decentralized markov decision processes. In J. S. Rosenschein, T. Sandholm, M. Wooldridge, and M. Yokoo, editors, *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2003)*, pages 41–48, Melbourne, Australia, 2003.

T. J. M. Bench-Capon. Truth and consequence: Complementing logic with value in legal reasoning. *Information and Communications Technology Law*, 10(1), 2001.

J. Bentahar, B. Moulin, J. C. Meyer, and B. Chaib-draa. Bargaining with information. In C. Sierra, L. Sonenberg, N. Jennings, and M. Tambe, editors, *Proceedings of the third Intl. Conference on Autonomous Agents and Multi-Agent Systems*, pages 790–798, 2004.

T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284 (5):34–43, May 2001.

A. Birk. Boosting cooperation by evolving trust. *Applied Artificial Intelligence*, 14(8): 769–784, 2000.

A. Birk. Learning to trust. In R. Falcone, M. Singh, and Y.-H. Tan, editors, *Trust in Cyber-societies*, pages 133–144. Springer-Verlag, Berlin Heidelberg, 2001.

A. Biswas, S. Sen, and S. Debnath. Limiting deception in groups of social agents. *Applied Artificial Intelligence Journal (Special issue on "Deception, Fraud and Trust in Agent Societies")*, 14(8):785–797, 2000.

G. Boella, J. Hulstijn, and L. van der Torre. Persuasion strategies in dialogue. In *Proceedings of CMNA Workshop at ECAI'04, Valencia*, 2004.

S. Brainov and T. Sandholm. Contracting with uncertain level of trust. In *Proceedings of the 1st ACM conference on Electronic commerce*, pages 15–21. ACM Press, 1999. ISBN 1-58113-176-3. doi: http://doi.acm.org/10.1145/336992.336998.

F. Brandt. Cryptographic protocols for secure second-price auctions. In M. Klush and F. Zambonelli, editors, *Lecture Notes in Artificial Intelligence*, volume 2812, pages 154–165. Berlin et al., 2001.

F. Brandt. A verifiable bidder-resolved auction protocol. In *Workshop on Deception, Fraud and Trust in Agent Societies, AAMAS 2002, Bologna, Italy*, pages 18–25. 2002.

R. Burt. *Toward a structural theory of action. Network models of social structure, perception, and action.* Academic Press, New York, NY, 1982.

L. Busch and I. J. Hortsmann. Endogenous incomplete contracts: a bargaining approach. *Canadian Journal of Economics*, 32(4), 1999.

V. Buskens. The social structure of trust. *Social Networks*, 20:265–298, 1998.

T. Calvo, A. Kolesarova, M. Komornikova, and R. Mesiar. Aggregation operators: Properties, classes and construction methods. In T. Calvo, G. Mayor, and R. Mesiar, editors, *Studies in Fuzziness and Soft Computing, Vol. 97*, pages 3–104. 2002.

K. Carley. A theory of group stability. *American Sociological Review*, 56:331–354, 1991.

C. Castelfranchi and R. Falcone. Principles of trust for MAS: cognitive anatomy, social importance, and quantification. In *Proceedings of the International Conference of Multi-Agent Systems (ICMAS'98)*, pages 72–79, 1998.

C. Castelfranchi and R. Falcone. Social trust: A cognitive approach. In C. Castelfranchi and Y.-H. Tan, editors, *Trust and Deception in Virtual Societies*, pages 55–90. Kluwer Academic Publishers, 2000a.

C. Castelfranchi and R. Falcone. Trust is much more than subjective probability: Mental components and sources of trust. In *Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS 33 CDROM) (online edition)*, volume 6. IEEE (online publication), 2000b.

K. M. Chandy, A. Fuchs, B. Janssen, D. Mulchandani, and M. Weiser. Ic online: Ubiquitous computing: The future of development? *IEEE Distributed Systems Online*, 3 (3), 2002.

H. H. Clark. *Using Language.* Cambridge University Press, Cambridge, England, 1996.

M. Colombetti. A commitment-based approach to agent speech acts and conversations. In *Proceedings of the Workshop on Agent Languages and Conversational Policies*, pages 21–29, 2000.

V. Conitzer and T. Sandholm. Revenue failures and collusion in combinatorial auctions and exchanges with vcg payments. In *6th Workshop on Agent Mediated Electronic Commerce (AMEC-04), New York, NY, USA*, page ??, 2004.

R. Conte and C. Castelfranchi. From conventions to prescriptions - towards an integrated view of norms. *Artificial Intelligence and Law*, 4(7):323–340, 1999.

P. Dasgupta. Trust as a commodity. In D. Gambetta, editor, *Trust: Making and Breaking Cooperative Relations*, pages 49–72. Blackwell, 1998.

P. Dasgupta and E. Maskin. Efficient auctions. *Quarterly Journal of Economics*, 115: 341–388, 2000.

R. K. Dash, D. C. Parkes, and N. R. Jennings. Computational mechanism design: A call to arms. *IEEE Intelligent Systems*, 18(6):40–47, 2003.

R. K. Dash, S. D. Ramchurn, and N. R. Jennings. Trust-based mechanism design. In *Proceedings of the Third International Conference on Autonomous Agents and Multi-Agent Systems, New York*, volume 2, pages 726–753, 2004.

C. Dellarocas. Towards incentive-compatible reputation management. In *Workshop on Deception, Fraud and Trust in Agent Societies, Bologna, Italy*, pages 26–40, 2002.

F. Dignum, B. Dunin-Kęplicz, and R. Berbrugge. Agent theory for team formation by dialogue. In C. Castelfranchi and Y. Lespérance, editors, *Intelligent Agents VII: Proceedings of the 7th International Workshop on Agent Theories, Architectures, and Languages (ATAL-2000)*, volume 1986 of *Lecture Notes in Computer Science*, pages 150–166. Springer Verlag, Berlin, Germany, 2000.

P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-persons games. *Artificial Intelligence*, 77(2):321–358, 1995.

eBay. http://www.ebay.com, 2003.

M. Elvang-Gøransson, P. Krause, and J. Fox. Acceptability of arguments as 'logical uncertainty'. In M. Clarke, R. Kruse, and S. Moral, editors, *Proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU 1993)*, number 747 in Lecture Notes in Computer Science, pages 85–90. Springer, 1993a.

M. Elvang-Gøransson, P. Krause, and J. Fox. Dialectic reasoning with inconsistent information. In D. Heckerman and A. Mamdani, editors, *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, pages 114–121, Washington D. C., USA, 1993b. Morgan Kaufmann.

J. C. Ely, D. Fudenberg, and D. K. Levine. When is reputation bad? (discussion paper no. 2035). Technical report, Harvard Institute of Economic Research, May 2004.

M. Esteva, J. A. Rodríguez, C. Sierra, P. Garcia, and J. L. Arcos. On the formal specifications of electronic institutions. *Lecture Notes in Artificial Intelligence*, 1991:

126–147, 2001. F. Dignum and C. Sierra editors, Agent-mediated Electronic Commerce (The European AgentLink Perspective).

P. Faratin, C. Sierra, and N. R. Jennings. Negotiation decision functions for autonomous agents. *International Journal of Robotics and Autonomous Systems*, 24(3-4):159–182, 1998.

P. Faratin, C. Sierra, and N. R. Jennings. Using similarity criteria to make trade-offs in automated negotiations. *Artificial Intelligence*, 142(2):205–237, 2002.

S. Fatima, M. Wooldridge, and N. R. Jennings. Optimal negotiation strategies for agents with incomplete information. In J.-J. Meyer and M. Tambe, editors, *Intelligent Agent series VIII: Proceedings of the 8th International Workshop on Agent Theories, Architectures, and Languages (ATAL 2001)*, volume 2333 of *Lecture Notes in Computer Science*, pages 53–68. Springer, 2001.

S. Fatima, M. Wooldridge, and N. R. Jennings. Multi-issue negotiation under time constraints. In C. Castelfranchi and L. Johnson, editors, *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2002)*, pages 143–150. ACM Press, 2002.

S. S. Fatima, M. Wooldridge, and N. R. Jennings. An agenda-based framework for multi-issue negotiation. *Artificial Intelligence*, 152(1):1–45, 2004.

M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica. Free-riding and whitewashing in peer-to-peer systems. In *Proceedings of the ACM SIGCOMM workshop on Practice and theory of incentives in networked systems*, pages 228–236. ACM Press, 2004. ISBN 1-58113-942-9. doi: http://doi.acm.org/10.1145/1016527.1016539.

FIPA. Communicative Act Library Specification. Technical Report XC00037H, Foundation for Intelligent Physical Agents, 10 August 2001.

FIPA. Abstract architecture specification, version j. http://www.fipa.org/repository, 2002.

R. Fisher and W. Ury. *Getting to Yes: Negotiating Agreement Without Giving In.* Penguin Books, New York, NY 10014, USA, 1983.

N. Fornara and M. Colombetti. Defining interaction protocols using a commitment-based agent communication language. In J. S. Rosenschein, T. Sandholm, M. Wooldridge, and M. Yokoo, editors, *Proc. of the 2nd International Joint Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 520–527, 2003.

I. Foster and C. Kesselman, editors. *The Grid, Blueprint for a New Computing Infrastructure.* Morgan Kaufmann Inc, 1998.

J. Fox and S. Parsons. Arguing about beliefs and actions. In A. Hunter and S. Parsons, editors, *Applications of Uncertainty Formalisms*, number 1455 in LNCS, pages 266–302. Springer, 1998.

D. Gambetta. Can we trust trust? In D. Gambetta, editor, *Trust: Making and Breaking Cooperative Relations*, pages 213–237. Blackwell, 1998.

G. Gans, M. Jarke, S. Kethers, and G. Lakemeyer. Continuous requirements management for organisation networks: a (dis-)trust-based approach. *Requirements Engineering Journal*, 8(1):4–22, 2003.

A. Giovannucci, J. A. Rodriguez-Aguilar, A. Reyes, F. X. Noria, and J. Cerquides. Towards automated procurement via agent-aware negotiation support. In C. Sierra, L. Sonenberg, N. Jennings, and M. Tambe, editors, *Proceedings of the third Intl. Conference on Autonomous Agents and Multi-Agent Systems*, pages 244–252, 2004.

T. Grandison and M. Sloman. A survey of trust in internet applications. *IEEE Communications Surveys and Tutorials*, 4(4):2–16, 2000.

C. L. Hamblin. *Fallacies.* Methuen, London, 1970.

D. Harel. Dynamic logic. In D. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic Volume II*, pages 497–604. D. Reidel Publishing Company, 1984.

M. He, N. R. Jennings, and H. Leung. On agent-mediated electronic commerce. *IEEE Trans on Knowledge and Data Engineering*, 15(4):985–1003, 2003.

A. Herzberg, Y. Mass, J. Michaeli, D. Naor, and Y. Ravid. Access control meets public key infrastructure, or: Assigning roles to strangers. In *IEEE Symposium on Security and Privacy, Oakland*, pages 2–14. IEEE Computer Society (online publication), May 2000.

D. Hitchcock. Aristotle's theory of argument evaluation. In K. Boudouris and J. Poulakos, editors, *The Philosophy of Communication*, volume 1. Ionia Press, Athens, Greece, 2002.

J. Hovi. *Games, Threats, and Treaties – Understanding commitments in international relations.* Pinter, 1998.

M. Hsu and V. Soo. A secure multi-agent Vickrey auction scheme. In *Workshop on Deception, Fraud and Trust in Agent Societies, AAMAS 2002, Bologna, Italy*, pages 86–91, 2002.

T. D. Huynh, N. R. Jennings, and N. Shadbolt. Fire: An integrated trust and reputation model for open multi-agent systems. In *Proceedings of the 16th European Conference on Artificial Intelligence , Valencia , Spain*, pages 18–22, 2004.

N. Islam. From smart to autonomous phones. *IEEE pervasive computing*, 3(3), July-September 2004.

P. Jehiel and B. Moldovanu. Efficient design with interdependent valuations. *Econometrica*, 69(5):1237–59, 2001.

N. R. Jennings. Commitments and conventions: The foundation of coordination in multi-agent systems. *The Knowledge Engineering Review*, 8(3):223–250, 1993.

N. R. Jennings. On agent-based software engineering. *Artificial Intelligence*, 117(2): 277–296, 2000.

N. R. Jennings. An agent-based approach for building complex software systems. *Communications. of the ACM*, 44(4):35–41, 2001.

N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge. Automated negotiation: prospects, methods and challenges. *International Journal of Group Decision and Negotiation*, 10(2):199–215, 2001.

N. R. Jennings, P. Faratin, T. J. Norman, P. O'Brien, and B. Odgers. Autonomous agents for business process management. *International Journal of Applied Artificial Intelligence*, 14(2):145–189, 2000.

N. R. Jennings, S. Parsons, P. Noriega, and C. Sierra. On argumentation-based negotiation. In *Proceedings of the International Workshop on Multi-Agent Systems*, Boston, USA, 1998.

A. Josang and R. Ismail. The beta reputation system. In *Proceedings of the 15th Bled Conference on Electronic Commerce, Bled, Slovenia*, 2002.

R. Jurca and B. Faltings. An incentive compatible reputation mechanism. In *Proceedings of the IEEE Conference on E-Commerce CEC03*, pages 285–292, 2003a.

R. Jurca and B. Faltings. Towards incentive-compatible reputation management. In R. Falcone, S. Barber, L. Korba, and M. Singh, editors, *Lecture Notes in Computer Science*, volume 2631, pages 138–147. 2003b.

R. Jurca and B. Faltings. Confess: Eliciting honest feedback without independent verification authorities. In *Sixth International Workshop on Agent Mediated Electronic Commerce (AMEC VI 2004), New York*, page ??, 2004.

A. Kakas and P. Moraitis. Argumentation based decision making for autonomous agents (to appear). In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2003)*, pages 883–890, Melbourne, Australia, 2003.

M. Karlins and H. I. Abelson. *Persuasion: How Opinions and Attitudes are Changed.* Springer, second edition, 1970.

J. O. Kephart and D. M. Chess. The vision of autonomic computing. *IEEE Computer*, 36(1):41–50, 2003.

E. Kersten and G. Lo. Negotiation support systems and software agents in e-business negotiations. In *Proceedings of the 1st International Conference on E-Business*, 2001.

M. Klein, P. Faratin, H. Sayama, and Y. Bar-Yam. Protocols for negotiating complex contracts. *IEEE Intelligent Systems*, 18(6):32–38, 2003.

P. Kollock. The production of trust in online markets. *Advances in Group Processes*, 16:99–124, 1999.

S. Kraus, K. Sycara, and A. Evenchik. Reaching agreements through argumentation: A logical model and implementation. *Artificial Intelligence*, 104(1–2):1–69, 1998.

P. Krause, S. Ambler, M. Elvang-Gøransson, and J. Fox. A logic of argumentation for reasoning under uncertainty. *Computational Intelligence*, 11:113–131, 1995.

V. Krishna. *Auction Theory*. Academic Press, San Diego, 2002.

Y. Labrou, T. Finin, and Y. Peng. Agent communication languages: The current landscape. *Intelligent Systems*, 14(2):45–52, 1999.

C.-J. Liau. Belief, information acquisition, and trust in multi-agent systems: a modal logic formulation. *Artif. Intell.*, 149(1):31–60, 2003. ISSN 0004-3702. doi: http://dx.doi.org/10.1016/S0004-3702(03)00063-8.

G. L. Lilien, P. Kotler, and S. K. Moorthy. *Marketing Models*. Prentice-Hall, USA, 1992.

F. López y López, M. Luck, and M. d'Inverno. Constraining autonomy through norms. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 674–681, 2002.

E. H. Mamdani. Applications of fuzzy logic to approximate reasoning using linguistic synthesis. *IEEE Transactions on Computers*, 26(12):1182–1191, 1977.

J. Manyika and H. Durrant-Whyte. *Data Fusion and Sensor Management: A Decentralized Information-Theoretic Approach*. Ellis Horwood, 1997.

S. Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, Department of Mathematics and Computer Science, University of Stirling, 1994.

M. Marx and J. Treur. Trust dynamics formalised in temporal logic. In Y. Z. L. Chen, editor, *Proc. of the Third International Conference on Cognitive Science, ICCS 2001*, pages 359–363. USTC Press, Beijing, 2001.

A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic Theory*. Oxford University Press, New York, 1995.

Y. Mass and O. Shehory. Distributed trust in open multi-agent systems. In R. Falcone, M. Singh, and Y.-H. Tan, editors, *Trust in Cyber-societies*, pages 159–173. Springer-Verlag, Berlin Heidelberg, 2001.

N. Maudet and B. Chaib-draa. Commitment-based and dialogue-game based protocols–news trends in agent communication language. *Knowledge Engineering Review*, 17(2): 157–179, 2003.

J. Mayfield, Y. Labrou, and T. Finin. Evaluating kqml as an agent communication language. In M. J. Wooldridge, J. P. Müller, and M. Tambe, editors, *Intelligent Agents II*, volume 1039 of *Lecture Notes in Computer Science*, pages 347–360. Springer Verlag, Berlin, Germany, 1996.

P. McBurney, S. Parsons, and M. Wooldridge. Desiderata for agent argumentation protocols. In C. Castelfranchi and L. Johnson, editors, *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2002)*, pages 402–409, New York, USA, 2002. ACM Press.

P. McBurney, R. M. van Eijk, S. Parsons, and L. Amgoud. A dialogue-game protocol for agent purchase negotiations. *Journal of Autonomous Agents and Multi-Agent Systems*, 7(3):235–273, 2003.

D. McFarlane. Interruption of people in humancomputer interaction: A general unifying definition of human interruption and taxonomy. Technical Report 5510–97-9870, Naval Research Laboratory Report, 1997.

D. McFarlane. Coordinating the interruption of people in human-computer interaction. *HumanComputer Interaction - INTERACT'99*, pages 295–303, 1999.

R. Mirrlees. An exploration in the theory of optimum income taxation. *Review of Economic Studies*, 38:175–208, 1971.

L. D. Molm, N. Takahashi, and G. Peterson. Risk and trust in social exchange: An experimental test of a classical proposition. *American Journal of Sociology*, 105:1396–1427, 2000.

L. Mui, M. Mohtashemi, and A. Halberstadt. A computational model of trust and reputation for e-business. In *35th Hawaii International Conference on System Science (HICSS 35 CDROM)*. IEEE (online publication), 2002.

R. Mukherjee, B. Banerjee, and S. Sen. Learning mutual trust. In R. Falcone, M. Singh, and Y.-H. Tan, editors, *Trust in Cyber-societies*, pages 145–158. Springer-Verlag, Berlin Heidelberg, 2001.

A. Muthoo. *Bargaining Theory with Applications.* Cambridge University Press, 1999.

J. F. Nash. Two-person cooperative games. *Econometrica*, 21:128–140, 1953.

M. J. Osborne and A. Rubinstein. *Bargaining and Markets.* Academic Press Inc., 1990.

P. Panzarasa, N. Jennings, and T. Norman. Social mental shaping: Modelling the impact of sociality on the mental states of autonomous agents. *Computational Intelligence*, 4 (17):738–782, 2001.

S. Parsons, C. Sierra, and N. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.

S. Parsons, M. Wooldridge, and L. Amgoud. An analysis of formal inter-agent dialogues. In C. Castelfranchi and L. Johnson, editors, *Proceedings of the 1st international joint conference on Autonomous agents and multiagent systems (AAMAS 2002)*, pages 394–401. ACM Press, 2002. ISBN 1-58113-480-0. doi: http://doi.acm.org/10.1145/544741.544835.

S. Parsons, M. Wooldridge, and L. Amgoud. On the outcomes of formal inter-agent dialogues. In J. Rosenschein, T. Sandholm, M. Wooldridge, and M. Yokoo, editors, *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2003)*, pages 616–623. ACM Press, 2003.

A. Pekec and M. H. Rothkopf. Combinatorial auction design. *Management Science*, 49 (11):1485–1503, 2003.

R. Porter, A. Ronen, Y. Shoham, and M. Tennenholtz. Mechanism design with execution uncertainty. In *UAI-02*, 2002.

S. Poslad, M. Calisti, and P. Charlton. Specifying standard security mechanisms in multi-agent systems. In R. Falcone, S. Barber, L. Korba, and M. Singh, editors, *Lecture Notes in Computer Science*, volume 2631, pages 163–176. 2003.

H. Prakken and G. Sartor. The role of logic in computational models of legal argument: a criticial survey. In A. Kakas and F. Sadri, editors, *Computational Logic: From Logic Programming into the Future (In honour of Bob Kowalski)*, volume 2048 of *Lecture Notes in Computer Science*, pages 342–3. Springer Verlag, Berlin, Germany, 2001.

L. Prechelt. INCA: A multi-choice model of cooperation under restricted communication. *BioSystems*, 37(1-2):127–134, 1996.

M. Prietula. Advice, trust, and gossip among artificial agents. In A. Lomi and E. Larsen, editors, *Simulating Organisational Societies: Theories, Models and Ideas.* MIT Press, 2000.

D. Pynadath and M. Tambe. Multiagent teamwork: Analyzing key teamwork theories and models. In C. Castelfranchi and L. Johnson, editors, *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems*, volume 2, pages 873–880, 2002.

I. Rahwan, P. McBurney, and L. Sonenberg. Towards a theory of negotiation strategy (a preliminary report). In S. Parsons and P. Gmytrasiewicz, editors, *Proceedings of the 5th Workshop on Game Theoretic and Decision Theoretic Agents (GTDT-2003)*, pages 73–80, 2003a.

I. Rahwan, S. D. Ramchurn, N. R. Jennings, P. McBurney, S. D. Parsons, and L. Sonenberg. Argumentation-based negotiation. *The Knowledge Engineering Review*, 18 (4), 2003b.

I. Rahwan, L. Sonenberg, and F. Dignum. On interest-based negotiation. In F. Dignum, editor, *Advances in Agent Communication*, volume 2922 of *Lecture Notes in Artificial Intelligence*. Springer Verlag, Berlin, Germany, 2003c.

I. Rahwan, L. Sonenberg, and F. Dignum. Towards interest-based negotiation (to appear). In J. S. Rosenschein, T. Sandholm, M. Wooldridge, and M. Yokoo, editors, *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2003)*, pages 773–780, Melbourne, Australia, 2003d.

H. Raiffa. *The Art and Science of Negotiation*. Belknapp, 1982.

S. D. Ramchurn, B. Deitch, M. K. Thompson, D. C. de Roure, N. R. Jennings, and M. Luck. Minimising intrusiveness in pervasive computing environments using multi-agent negotiation. In *Proceedings of the 1st International Conference on Mobile and Ubiquitous Systems*, pages 364–372, 2004a.

S. D. Ramchurn, D. Huynh, and N. R. Jennings. Trust in multi-agent systems. *The Knowledge Engineering Review*, 19, 2004b.

S. D. Ramchurn, N. R. Jennings, and C. Sierra. Persuasive negotiation for autonomous agents: A rhetorical approach. In C. Reed, editor, *Workshop on the Computational Models of Natural Argument, IJCAI*, pages 9–18, 2003a.

S. D. Ramchurn, C. Sierra, L. Godo, and N. R. Jennings. A computational trust model for multi-agent interactions based on confidence and reputation. In R. Falcone, S. Barber, L. Korba, and M. Singh, editors, *Workshop on Deception, Trust, and Fraud, AAMAS*, pages 69–75, 2003b.

S. D. Ramchurn, C. Sierra, L. Godo, and N. R. Jennings. Credit: A computational trust model for multi-agent systems based on confidence and reputation. *Journal of Artificial Intelligence Research (JAIR) (under review)*, 2004c.

S. D. Ramchurn, C. Sierra, L. Godo, and N. R. Jennings. Devising a trust model using confidence and reputation. *Applied Artifical Intelligence*, 18(9-10), 2004d.

A. S. Rao and M. P. Georgeff. Formal models and decision procedures for multi-agent systems. Technical Report 61, Australian Artificial Intelligence Institute, 171 La Trobe Street, Melbourne, Australia, 1995.

P. Resnick and R. Zeckhauser. Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system. In M. R. Baye, editor, *Advances in Applied Microeconomics*, volume 11, pages 127–157. Elsevier Science, 2002.

M. Ripeanu, I. Foster, and A. Iamnitchi. Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. *IEEE Internet Computing special issue on Peer-to-Peer Networking*, 6(1):50–57, 2002.

J. Rosenschein and G. Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. MIT Press, Cambridge MA, USA, 1994.

J. C. Rubiera, J. M. M. Lopez, and J. D. Muro. A fuzzy model of reputation in multi-agent systems. In *Proceedings of the fifth international conference on Autonomous agents*, pages 25–26. ACM Press, 2001. ISBN 1-58113-326-X. doi: http://doi.acm.org/10.1145/375735.375815.

A. Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica*, 50:97–109, 1982.

J. Sabater and C. Sierra. REGRET: a reputation model for gregarious societies. In C. Castelfranchi and L. Johnson, editors, *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 475–482, 2002.

N. Sadeh. *M-Commerce: Technologies, Services, and Business models*. Wiley computer publishing, 2002.

F. Sadri, F. Toni, and P. Torroni. Dialogues for negotiation: agent varieties and dialogue sequences. In J.-J. Meyer and M. Tambe, editors, *Intelligent Agent Series VIII: Proceedings of the 8th International Workshop on Agent Theories, Architectures, and Languages (ATAL 2001)*, volume 2333 of *Lecture Notes in Computer Science*, pages 69–84. Springer, 2001a.

F. Sadri, F. Toni, and P. Torroni. Logic agents, dialogues and negotiation: an abductive approach. In K. Stathis and M. Schroeder, editors, *Proceedings of the AISB 2001 Symposium on Information Agents for E-Commerce*, 2001b.

F. Sadri, F. Toni, and P. Torroni. Abductive logic programming architecture for negotiating agents. In *Proceedings of the 8th European Conference on Logics in Artificial Intelligence (JELIA 2002)*, volume 2424 of *Lecture Notes in Computer Science*, pages 419–431. Springer, 2002.

T. Sandholm. Distributed rational decision making. In G. Weiss, editor, *Multi-Agent Systems: A Modern Approach To Distributed Artificial Intelligence*. The MIT Press, 1999.

T. Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135(1–2):1–54, 2002.

T. Sandholm. Making markets and democracy work: A story of incentives and computing. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1649–1671, 2003.

T. Sandholm and R. Crites. Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems*, 37:147–166, 1995. Special Issue on the Prisoner's Dilemma.

M. Satyanarayanan. Pervasive computing: Vision and challenges. *IEEE Personal Communications*, 2001.

L. Savage. *The foundations of Statistics*. John Wiley and Sons, 1954.

T. C. Schelling. *The Strategy of Conflict*. Oxford University Press, New York, 1963.

M. Schillo, P. Funk, and M. Rovatsos. Using trust for detecting deceptive agents in artificial societies. *Applied Artificial Intelligence, Special Issue on Trust, Deception, and Fraud in Agent Societies*, 14(8):825–848, 2000.

H. Schmeck, T. Ungerer, and L. C. Wolf, editors. *Trends in Network and Pervasive Computing - ARCS 2002, International Conference on Architecture of Computing Systems, Karlsruhe, Germany, April 8-12, 2002, Proceedings*, volume 2299 of *Lecture Notes in Computer Science*, 2002. Springer. ISBN 3-540-43409-7.

J. Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, New York, 1969.

S. Sen. Reciprocity: a foundational principle for promoting cooperative behavior among self-interested agents. In *Proceedings of the 2nd International Conference on Multi-Agent Systems*, pages 322–329. AAAI Press, Menlo Park, CA, 1996.

S. Sen, A. Biswas, and S. Debnath. Believing others : Pros and cons. In *Proceedings of the International Conference on Multi-Agent Systems*, pages 279–286, 2000.

S. Sen and P. S. Dutta. The evolution and stability of cooperative traits. In C. Castelfranchi and L. Johnson, editors, *Proceedings of the 1st Intenational Joint Conference on Autonomous Agents and Multiagent Systems*, volume 3, pages 1114–1120, 2002.

S. Sen and N. Sajja. Robustness of reputation-based trust: Boolean case. In C. Castelfranchi and L. Johnson, editors, *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems*, volume 1, pages 288–293, 2002.

M. Seth. Introduction to web services management. http://www.developer.com/services/article.php/1583511, 2003. Developer.com.

C. Sierra, N. R. Jennings, P. Noriega, and S. Parsons. A framework for argumentation-based negotiation. In M. Singh, A. Rao, and M. Wooldridge, editors, *Intelligent Agent IV: 4th International Workshop on Agent Theories, Architectures and Languages (ATAL 1997)*, volume 1365 of *Lecture Notes in Computer Science*, pages 177–192. Springer, 1998.

H. A. Simon. *The Sciences of the Artificial.* MIT Press, third edition, 1996.

M. P. Singh. A social semantics of agent communication lauguages. In F. Dignum and M. Greaves, editors, *Issues in Agent Communication*, volume 1916 of *Lecture Notes in Computer Science*, pages 31–45. Springer Verlag, Berlin, Germany, 2000.

M. P. Singh, B. Yu, and M. Venkatraman. Community-based service location. *Communications of the ACM*, 44(4):49–54, 2001.

A. M. Spence. *Market Signaling: Informational Transfer in Hiring and Related Screening Processes.* Harvard Economic Studies. Harvard University Press, Cambridge MA, USA, 1974.

Y. Tan and W. Thoen. Formal aspects of a generic model of trust for electronic commerce. In *Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS 33 CDROM).* IEEE (online publication), 2000.

P. Torroni. A study on the termination of negotiation dialogues. In C. Castelfranchi and L. Johnson, editors, *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2002)*, pages 1223–1230. ACM Press, 2002.

P. Torroni and F. Toni. Extending a logic based one-to-one negotiation framework to one-to-many negotiation. In P. P. A. Omicini and R. Tolksdorf, editors, *2nd International Workshop on Engineering Societies in the Agents World II (ESAW 2001)*, volume 2203 of *Lecture Notes in Computer Science*, pages 105–118. Springer, 2001.

D. Traum. Speech acts for dialogue agents. In M. Wooldridge and A. Rao, editors, *Foundations And Theories Of Rational Agents*, pages 169–201. Kluwer Academic, 1999.

G. Tsebelis. Are sanctions effective? a game theoretic analysis. *Journal of Conflict Resolution*, 34:3–28, 1990.

A. Tveit. Peer-to-peer based recommendations for mobile commerce. In *Proceedings of the 1st international workshop on Mobile commerce*, pages 26–29. ACM Press, 2001. ISBN 1-58113-376-6. doi: http://doi.acm.org/10.1145/381461.381466.

J. von Neuman and O. Morgenstern. *The Theory of Games and Economic Behaviour.* Princeton University Press, Princeton NJ, USA, 1944.

N. Vulkan. Economic implications of agent technology and e-commerce. *The Economic Journal*, 109(453):67–90, 1999.

D. N. Walton and E. C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning.* SUNY Press, Albany, NY, USA, 1995.

Y. Wang and J. Vassileva. Bayesian networks-based trust model in peer-to-peer networks. In *Proceedings of the 6th workshop on Trust, privacy, deception and fraud in agent systems*, pages 92–100, 2003.

A. Whitby, A. Jsang, and J.Indulska. Filtering out unfair ratings in bayesian reputation systems. In *Proceedings of the Workshop on Trust in Agent Societies, New York*, page ??, 2004.

P. Winoto, G. McCalla, and J. Vassileva. Non-monotonic offers bargaining protocol. In C. Sierra, L. Sonenberg, N. Jennings, and M. Tambe, editors, *Proceedings of the third Intl. Conference on Autonomous Agents and Multi-Agent Systems*, pages 1070–1078, 2004.

M. Witkowski, A. Artikis, and J. Pitt. Experiments in building experiential trust in a society of objective-trust based agents. In R. Falcone, M. Singh, and Y.-H. Tan, editors, *Trust in Cyber-societies*, pages 111–132. Springer-Verlag, Berlin Heidelberg, 2001.

M. Wooldridge. *An Introduction to MultiAgent Systems.* John Wiley and Sons, Chichester, England, 2002.

M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115–152, 1995.

M. Wooldridge and S. Parsons. Languages for negotiation. In W. Horn, editor, *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI-2000).* John Wiley, 2000.

D. J. Wu and Y. Sun. The emergence of trust in multi-agent bidding: A computational approach. In *Proceedings of the 34th Hawaii International Conference on System Sciences (HICSS 34 CD ROM)*, volume 1. IEEE (online publication), 2001.

R. R. Yager, J. Kacprzyk, and M. Fedrizzi. *Advances in the Dempster-Shafter Theory of Evidence.* Wiley, 1994.

T. Yamagishi, K. Cook, and M. Watabe. Uncertainty, trust, and commitment formation in the united states and japan. *American Journal of Sociology*, 104:165–94, 1998.

M. Yokoo and K. Hirayama. Algorithms for distributed constraint satisfaction: A review. *Autonomous Agents and Multi-Agent Systems*, 3(2):198–212, 2000.

B. Yu and M. P. Singh. Distributed reputation management for electronic commerce. *Computational Intelligence*, 18(4):535–549, 2002a.

B. Yu and M. P. Singh. An evidential model of reputation management. In C. Castelfranchi and L. Johnson, editors, *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems*, volume 1, pages 295–300, 2002b.

B. Yu and M. P. Singh. Searching social networks. In J. S. Rosenschein, T. Sandholm, M. Wooldridge, and M. Yokoo, editors, *Proceedings of the 2nd International Joint Conference on Autonomous and Multi-Agent Systems*, pages 65–72, 2003.

G. Zacharia and P. Maes. Trust through reputation mechanisms. *Applied Artificial Intelligence*, 14:881–907, 2000.

L. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

R. Zeckhauser and W. K. Viscusi. Risk within reason. *Science*, 248(4955):559–564, 1990.

D. Zeng and K. Sycara. Benefits of learning in negotiation. In *Proceedings of the 14th National Conference on Artificial Intelligence and 9th Innovative Applications of Artificial Intelligence Conference (AAAI-97/IAAI-97)*, pages 36–42, Menlo Park, USA, 27–31 1997. AAAI Press.

Appendix A

Trust in Practice

We choose the semantic web to illustrate the practical applications of trust for open multi-agent systems. This is because, while potential applications of agent based systems such as ubiquitous computing and pervasive computing applications are still in their infancy, the semantic web is building upon the considerable success of the world wide web and technologies associated with it. Moreover, the semantic web is strongly motivated by concepts in multi-agent systems (e.g. reasoning under uncertainty, ontologies, communication languages). It can therefore be considered that the semantic web will provide the testbed for the first large scale application of agent-based systems in every day life. For these reasons, we provide the following vision of the semantic web (adapted from (Berners-Lee et al., 2001)) and detail the roles of trust models and interaction mechanisms within it.

Lucy and Peter have to organise a series of appointments to take their mother to the doctor for a series of physical therapy sessions. (We identify the need for trust at each step of the scenario in *italics*).

At the doctor's office, Lucy instructed her Semantic Web agent through her handheld Web browser. The agent promptly retrieved information about Mom's prescribed treatment from the doctor's agent, looked up several lists of providers, and checked for the ones in-plan for Mom's insurance within a 20-mile radius of her home and with a rating of excellent or very good on *trusted* rating services.

> *The first interaction between Lucy's agent and the doctor's agent should involve a secure authentication protocol (see section 3.2.3) that would ensure that Lucy's agent is allowed to handle her mom's data. This protocol would first verify the true identity of Lucy's agent and assign to it the proper rights to handle the data. Also, the trusted rating services could be based on*

reputation mechanisms (see section 3.2.2). These reputation mechanisms could publish the ratings of health care providers and reward agents which return ratings with discounts on treatment costs to be paid to the advertised providers. This would make the mechanism incentive-compatible. Also, different providers could bid, via a trusted mechanism such as a secure Vickrey auction, to provide the requested service to Lucy's agent (see section 3.2.1). Provider agents would need to bid their true valuation of the treatment plan requested to win the bid whereas Lucy's agent would act as the auctioneer in this case.

Lucy's agent then began trying to find a match between available appointment times (supplied by the agents of individual providers through their Web sites) and Pete's and Lucy's busy schedules. In a few minutes the agent presented them with a plan. Pete didn't like it: University Hospital was all the way across town from Mom's place, and he would be driving back in the middle of rush hour. He set his own agent to redo the search with stricter preferences about location and time. Lucy's agent, having complete *trust* in Pete's agent in the context of the present task, automatically assisted by supplying access certificates and shortcuts to the data it had already sorted through.

The interaction between individual providers and the user agents (Lucy's and Pete's) needs a secure mechanism that ensures messages transmitted between all parties are not manipulated. Pete's agent could enhance the search for trustworthy potential providers by looking at its past interaction history with them (see section 3.1.1) rather than looking at only the reputed ones (see sections 3.1.2 and 3.2.2). It could also use referrals of other agents in the network to get in touch with a trustworthy agent it does not directly know.

Almost instantly the new plan was presented: a much closer clinic and earlier times but there were two warning notes. First, Pete would have to reschedule a couple of his less important appointments. He checked that they were not a problem. The other was something about the insurance company's list failing to include this provider under physical therapists: "Service type and insurance plan status securely verified by other means," the agent reassured him. "(Details?)" .

Here the issue of reputation and distributed security is again raised (sections 3.1.2 and 3.2.3). The 'other means' that have helped to check the validity of the insurance company may pertain to an analysis of the certificates it *provided that linked it to trusted sources. These certificates could provide evidence of the provider's compliance with laws and regulations of the country or certain quality standards that are equivalent to those needed by the insurance company.*

Lucy registered her assent at about the same moment Pete was muttering, "Spare me the details," and it was all set. (Of course, Pete couldn't resist the details and later that night had his agent explain how it had found that provider even though it was not on the proper list.)

Here, the need for an agent to demonstrate how it could flexibly deal with different beliefs it acquired in the environment about potential interaction partners is highlighted (see section 3.1.3). This implies a higher level reasoning ability than just an evaluation reputation of providers for example. The agent should also be able to reason about the selected provider's location and treatment facilities to decide on whether to trust that provider in being able to supply the required services.

Appendix B

Using CREDIT in a Bandwidth Trading Scenario

We consider an SU agent α trying to find a reliable SP agent in its environment in order to get a good internet connection to perform voice over IP and access web services for a reasonable price. Figure B.1 graphically summarises the different steps involved in using CREDIT during the interaction between α and a given SP agent. Thus, agent α first selects the issues it intends to contract an SP agent for (see block 1 in figure B.1) and then queries other agents in the environment, asking them how they rate the available SP agents (see block 2). We assume agent α has also interacted with some of these SP agents in the past and has built up a history of interactions with them. From this history it has built up confidence values in each of the issues it wants to contract, given the context, as shown in section 5.2.3.2.

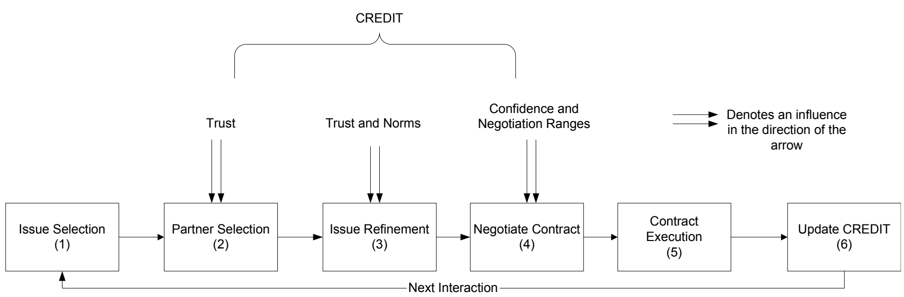


FIGURE B.1: Using and updating CREDIT in interactions.

The reputation values for each SP agent are fed into CREDIT and combined with any available confidence values in order to compute the overall trust of each SP agent for each of the issues α wants to negotiate (using equations 5.4, 5.6 and 5.8). The fuzzy sets used to characterise performance of an agent per issue were those shown in figure 5.1 and these are transfered to the domain space of each issue using the procedure outlined in equation 5.7. Thus, assume four SP agents are found with the following overall trust ratings (see block 2), using equation 5.10, $T(\alpha, \beta_1, X(O)) = 0.8, T(\alpha, \beta_2, X(O)) = 0.4, T(\alpha, \beta_3, X(O)) = 0.6$, and $T(\alpha, \beta_4, X(O)) = 0.1$. From these measures, α therefore decides to choose β_1 as the interaction partner since it is the most trusted of all SP agents (given the issues α wants to contract and the weights these issues take in its utility function).

Having decided to choose agent β_1, α checks if the SU's rule in table 5.4 applies and how far β can be trusted on the premises of the rule (see block 3 in figure B.1). Having found that β is highly trustworthy on price and size (e.g. $T(\alpha, \beta_1, c) > 0.9$ *and* $T(\alpha, \beta_1, s) > 0.95$) α decides not to include the *qos* in the set of issues to be negotiated as discussed in section 5.3.2.2. Agent α then contacts β_1 to engage into negotiations.

Prior to negotiations, β_1, which is also using CREDIT in this case, checks if its trust in α is high enough to interact with it instead of other agents. Finding that α has a relatively high trust with respect to other agents (e.g. $T(\alpha, X(O)) = 0.8$), β_1 decides to interact with α but includes the *usage* issue in the number of issues to be contracted since α is not trusted on t_c and l (i.e. $T(\beta, \alpha, t_c) < 0.85$ *and* $T(\beta, \alpha, l) > 0.9$).

Thus the final set of issues to be negotiated by the two agents are: price, size, time of payment, security level, and usage. Given β_1's low confidence in α with respect to t_c and l, β_1 will shrink its negotiation range (see block 4), using the procedure described in section 5.3.2.1, from $[10, 20]$ days to $[10, 15]$ days for time of payment and shrink the negotiation range for the security requirement from $[5, 10]$ to $[7, 10]$. Similarly α's reduced confidence in β_1 on the issue of price will cause it to shrink its negotiation range on price from $[30, 80]$ to $[30, 40]$.

Having thus defined their negotiation ranges, the two agents will negotiate using their own negotiation strategies (see block 4). Thus, the two agents come to an agreed contract $O = \{c = 35, s = 3Mbits/s, l = 5, usage = 70, t_c = 14days\}$. While β_1 can defect from the agreement by demanding a higher price at a later time, and reducing the bandwidth allowed, α can defect by paying later than agreed and using the connection more than agreed (e.g. by sending spam, or using peer to peer programs). However, the SP agent, β_1, decides to play a P strategy at execution time in order to keep its reputation in the society high, while α decides to be N since it can find other suppliers if β_1 does not want to interact with it in the future (see block 5).

Therefore, β_1 achieves what has been agreed in the contract while α defects on all the issues that it controls. This means that α will pay the latest it can (i.e. 30 days instead of 14), defects from the level of security agreed by using unwarranted software (i.e. $l = 1$ instead of $l = 5$), and exceeds the number of connections allowed per second by using a peer-to-peer program (i.e. $usage = 100$ instead of $usage = 70$).

Once the bandwidth has been paid for and used by α, the two agents then analyse each other's performance of the agreed contract and update their trust in their counterparts (see block 6). Thus, α senses no utility loss on the part of β_1 on those issues which are not regimented by any institutional norms. Therefore α senses a lower probability of utility loss on these issues and this increases its confidence. Hence its trust in β_1 increases over each issue it handled correctly (given the procedure described in section 5.2.3.2). As a result α increases its overall trust in β_1 (e.g. say from $T(\alpha, \beta_1) = 0.88$ to $T(\alpha, \beta_1) = 0.90$). On the other hand, β_1 finds that it has incurred substantial utility loss on all issues that α handled in the contract. Using the procedure described in section 5.2.3.2, β_1 therefore decreases its confidence on all issues α handled and as a result reduces its overall trust in α (e.g. from $T(\beta_1, \alpha) = 0.8$ to $T(\beta_1, \alpha) = 0.70$).

The next time β_1 is contacted by α, β_1 might refuse any contract with it or else shrink its negotiation ranges so as to demand higher-valued contracts (for β_1) in order to compensate its past utility loss.